ISSN 0392-8896

UNA PUBBLICAZIONE DEL GRUPPO EDITORIALE JACKSON

PERSONAL SOFTWARE



Spedizione in abbonamento postale gruppo III/70

ANNO 4 N. 28
MAGGIO 1985 - L. 4.000

LA PRIMA RIVISTA EUROPEA DI SOFTWARE PER PERSONAL COMPUTER

GRENDEL PER SPECTRUM

SKY WALKER PER C 64

CACCIA ALLA COPPIA PER C 16

CACCIA AL 13 CON TI99

BIANCHI E NERI PER APPLE

GRAFICI DI FUNZIONI CON LO SHARP



In copertina: questo mese vi presentiamo Caccia alla coppia, un gioco di strategia per Apple.

PERSONAL SOFTWARE

N. 28
MAGGIO 1985

ARTICOLI

GUIDA

12 ROUTINE IN LINGUAGGIO MACCHINA di *Carlo Cappelli* — ZX81

37 CACCIA AL TREDICI di *Carlo Aceti* — TI99/4A

41 CACCIA ALLA COPPIA PER APPLE II, C 64, C 16 di *Claudio Poma* — Apple - C 64 - C 16

50 GRAFICI DI FUNZIONI TRIGONOMETRICHE di *Fabio Farina* e *Daniele Zampariolo* — Sharp

55 BIANCHI E NERI PER APPLE di *Stefano Guarinelli* — Apple

62 CMK 49: NOTE E MUSICA CON IL COMMODORE 64 — C 64

63 GRENDEL, IL VOSTRO RENUMBER di *Stefano Cerutti* — Spectrum

70 GOLLUM PER C 64 2° di *Stefano Guarinelli* — C 64

87 SKYWALKER PER C 64 di *Luca Marras* — C 64

93 ZERO E CROCI CON LO SPECTRUM di *Anacleto Furlan* — Spectrum

RUBRICHE

5 EDITORIALE di *Riccardo Paolillo*

7 POSTA

9 PERSONAL NEWS a cura di *Marco Giacobazzi*

I SEGRETI DEI PERSONAL:

101 APPUNTI DI UTILITA' CON TASTI FUNZIONE di *Alessandro Guida* — VIC 20 - C 64

104 COME (TENTARE DI) RECUPERARE PROGRAMMI BASIC di *Marcello Spero* — Spectrum

107 STRUTTURA INTERNA DELLE ISTRUZIONI BASIC di *Martino Sangiorgio* — Sharp

112 PICCOLI ANNUNCI

EDITORIALE

Il nostro medioevo

di Riccardo Paolillo

Quindici anni fa, nel 1970, Roberto Vacca scrisse un libro che conobbe un certo successo. Il titolo, *Il medioevo prossimo futuro*, e soprattutto il sottotitolo, la degradazione dei grandi sistemi, chiarivano immediatamente la tematica trattata.
L'autore, dopo aver preso in esame i maggiori sistemi sui quali si basa la società attuale (produzione, distribuzione di energia, trasporti, comunicazioni, trattamento delle informazioni, ecc.), ne metteva in luce il degrado. Più precisamente sosteneva che ogni sistema era soggetto ad una crescita disordinata destinata a portarlo ad una pericolosa situazione di complessità ed instabilità. Secondo l'autore, una concomitanza casuale di congestioni in molti sistemi nella stessa area avrebbe potuto causare un processo catastrofico destinato a paralizzare il funzionamento delle società più sviluppate e provocare la morte a milioni di persone.

Si trattava in definitiva di un saggio scritto da un pessimista, ma secondo una logica stringente anche se forse poco probabile.
Anche se ovviamente si tratta di un lavoro ormai datato (come ammise lo stesso autore in una postfazione scritta otto anni dopo, e per certi versi già superata anch'essa), sono rimasto abbastanza colpito rileggendo recentemente il capitolo dedicato ai calcolatori elettronici (il neologismo informatica era allora scarsamente usato). Intitolate emblematicamente *Speranze mal riposte e timori infondati dei calcolatori elettronici*, queste pagine scritte tanti anni fa contengono delle osservazioni sicuramente valide anche oggi nel campo dell'informatica personale.
L'autore afferma giustamente che per avere dei vantaggi dall'utilizzo del calcolatore, occorre prima definire la struttura e la logica del sistema esaminato, operare cioè un'analisi sistematica.
Questo tipo di approccio è richiesto anche nel caso dei calcolatori personali. Se, ad esempio si volessero memorizzare delle informazioni bisognerebbe ovviamente stabilirne il tipo e il formato. Per eseguire correttamente questa operazione, non basta elencare casualmente i dati su cui si intende lavorare; occorre esaminare il problema da un punto di vista più generale rispondendo a domande del tipo: che operazioni devo compiere sui dati, in che modo intendo ricercare le informazioni memorizzate, quali nuovi dati posso prevedere di aggiungere in futuro, quanti dati penso di memorizzare, e così via.
Un vantaggio indiretto, ma importantissimo dell'approccio sistematico, evidenziato anche da Roberto Vacca è costituito dalla possibilità di selezionare i dati evitando risultati superflui.
Infatti procedure poco curate dal punto di vista dell'analisi, tendono a produrre moli enormi di dati che anziché chiarire le idee, sortiscono l'effetto opposto. E' noto che la tendenza attuale, che privilegia l'immediatezza delle rappresentazioni grafiche, è quella di avere il minor numero possibile di dati significativi.



A parte l'indubbia competenza e sensibilità dell'autore, tra l'altro docente di Automazione del Calcolo e quindi appartenente ad un osservatorio privilegiato, resta il fatto che queste intuizioni, avute molto prima della nascita dei calcolatori personali, stupiscono un po'. Probabilmente (e direi fortunatamente) la velocità dell'evoluzione tecnologica è molto superiore all'adeguamento della mentalità umana. Questo significa che se in quindici anni cambia, e di molto, il panorama tecnico, in proporzione viene solo impercettibilmente modificato il nostro modo di pensare e operare.
Quindi niente paura: è ancora lontana la figura dell'uomo automa e viceversa quella della macchina umana!

SERVIZIO SOFTWARE

PERSONAL SOFTWARE



P.S. propone ai propri lettori i dischi o le cassette dei programmi pubblicati. I programmi, provati e garantiti, sono di immediato utilizzo.

P.S. n°	Programma	Sistema	Prezzo	Codice	Supporto
3	La carta del cielo Collisione	Apple II	30.000	1	Disco
4	Interi in precisione multipla Grafica 3D	Apple II	40.000	4	Disco
5	Pretty printer Shape table	Apple II	30.000	6	Disco
7	Data base modulare	Apple II	25.000	7	Disco
12-13	Wei-ch'i	CBM 3032	20.000	8	Cassetta
14	Tool-Kit	C 64	35.000	9	Cassetta
19	Type Writer	VIC 20	30.000	10	Disco
20	Scopa	C 64 - 3032	25.000	11	Cassetta

Per richiedere i programmi in contrassegno, pagando direttamente al postino la cifra indicata, inviare il seguente tagliando

Spedire in busta chiusa a Gruppo Editoriale Jackson - Via Rosellini, 12 - 20124 Milano

Inviatemi i seguenti nastri e/o dischi con i programmi pubblicati su P.S.



GRUPPO EDITORIALE JACKSON

Cod. ______ **a L.** ______

Cod. ______ **a L.** ______

Cod. ______ **a L.** ______

Cod. ______ **a L.** ______

Cod. ______ **a L.** ______

+ SPESE POSTALI (contributo fisso) L. 3.000 **TOTALE L.** ______

che pagherò al postino alla consegna del pacco.

Cognome ....

Nome ....

Indirizzo ....

CAP ....

Città ....

Firma ....

POSTA

Ancora sul software

Ho letto con interesse il Vs. editoriale sul n. 25 di **Personal Software** a proposito, appunto, del software.
Senza dubbio avete toccato vari punti importanti del problema e le cose che avete detto sono giuste, però...
Però credo che si possano esprimere anche altri punti di vista.
Io penso ad esempio che nessuno "pretenda" di avere del software gratis. Penso che chiunque, solo che abbia un minimo di raziocinio, solo che abbia provato a programmare in proprio e sappia perciò "di che lacrime grondi e di che sangue" il lavoro di programmazione, sia disposto a concedere che chi ha lavorato abbia il sacrosanto diritto di veder riconosciuto il proprio lavoro. Ma attenzione: anche lo scrivere un libro richiede mesi e mesi di faticoso lavoro e se è un libro tecnico richiede spesso ampia collaborazione di fotografi, disegnatori ecc.
Chi comprerebbe il libro se esso si presentasse in vetrina al prezzo di L. 495.600 (è il prezzo dell'Apple Works IIc, IVA compresa, per esempio).
E ce n'è di ben più costosi.
È chiaro che appena uscito in circolazione il primo, subito comparirebbero numerosissime le fotocopie!

Mettiamoci nei panni del rivenditore. Io non lo sono, ma se lo fossi non so proprio come mi comporterei, perché sarà pure disonesto dare al cliente un po' di software scopiazzato e gratuito, ma sarebbe altrettanto disonesto rifilargli una macchina da due, tre, quattro milioni senza un minimo di possibilità operativa e pretendere che il cliente, in aggiunta spenda un altro mucchio di soldi per farci qualcosa con quella sua macchina.
Posso esser d'accordo con Voi che se una azienda abbisogna di un programma speciale lo chieda e se lo paghi quanto conviene, così come pagherebbe e paga in effetti un consulente tecnico di alto livello. Ma per tutti gli altri, dagli hobbisti, agli studenti, agli insegnanti, che comprano con sacrificio il computer, ma che non lo usano per far soldi per cui la macchina "non si paga da sé", ebbene per questi ha senso mettere in vendita il software a prezzi proibitivi?
Ritorno al mio pensiero: nessuno si preoccupa di "proteggere" il proprio libro perché costa un prezzo talmente basso che nessuno si prende il disturbo di fotocopiarlo.
Un altro aspetto della faccenda: proprio per non mettere il venditore in primis nella irresistibile tentazione di smerciare software sottobanco, perché i costruttori non dotano le loro macchine almeno di un buon software di base? La Sinclair l'ha fatto con il QL.
Ho la sensazione che i softwaristi i guai se li cerchino.

Voi dite "...le persone che vendono o acquistano software copiato, sicuramente stupirebbero molto se un libraio offrisse loro la fotocopia di un libro piuttosto che una copia originale...", ma io dico che quegli stessi non si stupirebbero più se l'originale costasse mezzo milione o un milione o due milioni.

Senza contare poi che, mentre fra l'originale di un libro e la sua fotocopia, se non c'è una differenza sostanziale c'è almeno una differenza estetica per la diversa veste tipografica, la copia di un disco non si differenzia proprio per niente dall'originale, a parte forse l'etichetta. O ci sono dischetti rilegati in pelle e dischetti in brossura?
L'errore di fondo non starà forse nel fatto che la comparsa del computer ha suggerito la nascita di un nuovo mestiere che, forse, mestiere non è: quello del softwarista?
C'è sicuramente in giro qualche cervellone che pretende di viverci di rendita con questo mestiere. Tempo verrà che anche i softwaristi si rassegneranno, come i poeti, cervelloni anch'essi, che "carmina non dant panem".
Certo che per il rivenditore è considerato un fastidio vendere il software; il rivenditore ha già i guai suoi per dimostrare che dopo tutto sei milioni per una macchina sono giustificabilissimi, e per dimostrare che tale macchina può fare dieci volte di più e di meglio di quel che fa un C 64. Specialmente se il cliente ci vuol guardare dentro e si rende conto che...c'è appena il doppio di roba o tutt'al più tre volte tanto.

Discorso ozioso? Voi come rivista tecnica sareste in grado di dimostrare che l'Apple IIc vale più del doppio del complesso C 64 più drive 1541? E non parlo di quantità di componenti, che doppia non è, ma soltanto di capacità operativa. Cosa fa l'Apple IIc che il C 64 non possa fare? Ha il doppio di memoria (apparentemente), ma doppia memoria vuol dire, tout court, doppio prezzo, più un'aggiunta?
Si potrebbe continuare e ce ne sarebbe da dire.
Perdonatemi se ho voluto mettere il becco in queste faccende, ma la provocazione è venuta da Voi, dopo tutto.

Franco Braga
Treviglio (BG)



Ritorniamo volentieri sull'argomento software, sia per chiarire meglio il nostro pensiero e sia perché ci

PERSONAL SOFTWARE

POSTA

sembra doveroso rispondere al Sig. Braga del quale avevamo già ospitato una lettera nel numero 20 (Agosto/Settembre 84) di **Personal Software** *e che ci segue con una costanza di cui gli siamo grati.*

L'opinione del Sig. Braga è evidentemente (e giustamente) quella dell'utente finale, quindi di chi in definitiva deve sborsare grosse cifre per procurarsi del software. Eppure non capiamo come prima affermi che chi programma ha il diritto di vedere riconosciuto il proprio lavoro e poche righe dopo disconosca addirittura la figura del programmatore, definendolo un cervellone che vuol vivere di rendita.

Noi che siamo un po' programmatori e un po' acquirenti di software, riteniamo che chi programma lavora, e quindi abbia diritto a un giusto compenso: ne più e ne meno.

Il problema, in realtà viene dopo: se i costi di realizzazione del prodotto software possono essere suddivisi su un numero elevato di vendite, allora è giusto attendersi un prezzo di vendita estremamente basso. Ma se, come purtroppo spesso accade per effetto delle copie, il numero di vendite regolari è basso allora il prezzo di vendita aumenta notevolmente.

Siamo comunque d'accordo con lei che la politica vincente per tutti è sicuramente quella di produrre software applicativo da vendere a basso costo, e il mercato sembra attualmente orientato in questa direzione.

Anche la sua osservazione sul fatto di vendere sistemi completi di una dotazione di base di software ci trova d'accordo, anche se nel caso di home computer non sempre i programmi proposti sono quelli effettivamente desiderati dall'utente. Per quanto riguarda il parallelo libro-programma ci sembra che i termini in questione siano difficilmente paragonabili: sia perché i tempi di realizzazione sono diversi e sia per il differente livello di diffusione. In ogni caso non bisogna dimenticare che oltre ai libri in edizione economica, esistono anche quelli che costano come un buon programma!

Quanto al confronto finale tra C 64 e Apple IIc riteniamo che doppio di prezzo non significhi semplicemente doppio di prestazioni. Altrimenti un mainframe IBM dovrebbe essere in grado di pensare da solo data la differenza di prezzo con Apple e una Ferrari di volare se paragonata ad una Fiat 126!





TROVERETE NEL PROSSIMO NUMERO

IN EDICOLA DAL 26 MAGGIO

NUOVO TORNEO DI BRIDGE CON L'APPLE

MUSICA CON UN MSX

SUPER SPREADSHEET PER C 16

QUOZIENTE INTELLIGENZA PER SPECTRUM

RAM TEST PER C 64

ROUTINE 2 PER SPECTRUM

I CHING PER TI 99/4A

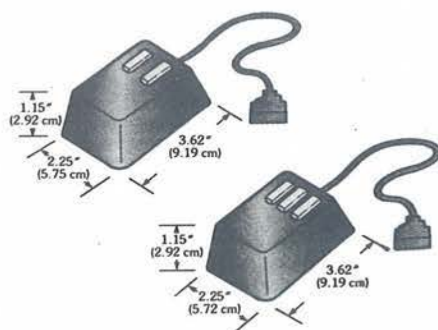
PROGETTO AEROMODELLO CON LO SPECTRUM

MONITOR PER C 64

TASTI FUNZIONE PER SPECTRUM

Mouse ottico dalla Shugart

La Corporation, ben introdotta nel settore dei drive per micro e minifloppy, ha deciso di aggredire anche il mercato delle appendici "manuali", offrendo agli OEM un mouse ottico. Si tratta di una scatoletta poco più grande del palmo di una mano, utilizzata per posizionare il cursore sullo schermo in un microsistema per scegliere funzioni, eseguire programmi e tracciare linee, reso famoso dall'uso privilegiato in alcune workstation avanzate e giunto al volgo con l'introduzione (ed il successo!) del Macintosh Apple. E' un apparecchio ottico, e offre pertanto alcuni vantaggi rispetto ad analoghi dispositivi meccanici. Non ha parti in movimento e quindi non necessita di manutenzione e pulizia periodica; è più leggero e silenzioso delle unità meccaniche; non avendo specchietti interni da mettere a punto, questo



mouse ottico Shugart può essere perfino più affidabile di altri prodotti similari. Le previsioni parlano chiaro: nel 1985 il mercato dei mouse sfiorerà i due milioni di unità; questo ne spiega l'interesse da parte di una casa già molto nota per altri prodotti. I modelli disponibili dispongono di due o tre bottoni ridefinibili dall'utente; per utilizzarli è necessario disporre sul tavolo una tavoletta di carta di riferimento (8 1/2 per 12 pollici) per il sistema di rilevamento del movimento, basato su un circuito appositamente realizzato e innescato da tre LED. Il mouse può essere mosso con una velocità di 25 pollici al secondo ed ha una "risoluzione" di 200 impulsi per pollice. E' fornito di interfaccia seriale, parallela o RS-232C.

Shugart Corporation
475 Dakmead Parkway
Sunnywale, CA 94086
Tel. 408-7374355

Logo in italiano dalla Commodore

Realizzata congiuntamente dalla Commodore Italiana e dalle Edizioni Elettroniche VIFI della Mondadori, con la supervisione di Giovanni Lariccia, ricercatore del CNR, la versione italiana di Logo per il C 64 viene distribuita su disco con una libreria di programmi ed un manuale in italiano, per 145.000 lire più IVA. Su Logo è stato impostato l'impegno didattico sulla programmazione all'interno dei progetti scuola varati da Commodore per l'anno scolastico che sta ormai volgendo al termine. Lucas e 100 scuole si stanno avviando a concludere il periodo di sperimentazione didattica nell'area della scuola dell'obbligo. Il primo progetto ha coinvolto 11 scuole elementari e medie di Milano e Provincia che sono state dotate di un laboratorio composto da 10 C 64 con 10 monitor a colori, 3 stampanti e 3 lettori di floppy disk, oltre a



cassette vergini e a software didattico: il già citato Logo ed altri sussidi per facilitare l'apprendimento del linguaggio BASIC. Gli obiettivi? Insegnare la programmazione strutturata in modo graduale, presentando piccoli programmi interattivi per far apprendere ai giovani un metodo di lavoro di validità generale, più che un singolo linguaggio di programmazione. Questo metodo, fondato sulla scomposizione di un problema in tanti piccoli sottoproblemi, è parso molto adeguato, sia dal punto di vista informatico, sia rispetto allo sviluppo generale della capacità di ragionare in modo sistematico. Il gruppo di coordinamento del progetto Lucas, sugli esiti del quale torneremo a sperimentazione conclusa, ritiene che l'insegnamento di questo metodo logico-analitico si inserisca a buon diritto nei programmi della scuola dell'obbligo per preparare i giovani ad affrontare con gli edeguati strumenti una società dominata dalla tecnologia. Scopo principale di questa nuova alfabetizzazione è rendere la persona cosciente dei processi che si svolgono dietro ad uno schermo o che guidano un robot, per non rinunciare agli spazi propri della libera creatività e fantasia dell'uomo. Nella stessa logica, con mezzi

ridotti rispetto a Lucas, si muove il progetto 100 scuole, che ha coinvolto altrettanti istituti di tutta Italia con l'obiettivo di iniziare gli alunni alla logica della programmazione usando la versione italiana di Logo alle elementari ed il BASIC nelle medie. I due progetti vengono seguiti da consulenti esperti nelle diverse materie e applicazioni. Tornando alla versione italiana di Logo, ci preme sottolineare, prima di concludere, che il lavoro svolto non è stato di mera traduzione, ma ha coinvolto anche aspetti di adattamento e revisione dell'interprete Logo su disco, delle sue varie procedure, dei programmi di utilità e degli esempi distribuiti sui due dischetti. Questi, ed il manuale, sono stati rielaborati e controllati da esperti di informatica applicata alla didattica e messi alla prova in ambienti didattici tradizionali, coinvolgendo scuole e docenti, per verificarne sul campo la reale validità.

Commodore Italiana S.p.A.
Via F.lli Gracchi, 48
20092 Cinisello Balsamo (MI)
Tel. 02-618321

Reclute e informatica

Perché non approffittare del tempo delle libere uscite per imparare o approfondire qualcosa di utile, anziché vagare smarriti per una città straniera? Tempo Reale, centro di cultura informatica della Magic Bus S.c.r.l. ha dato il via ad un'iniziativa rivolta ai militari di leva del Presidio bolognese. Per loro è infatti possibile iscriversi ad uno dei corsi organizzati da Tempo Reale beneficiando di uno sconto che si ottiene presentando alla segreteria il proprio tesserino militare. Gli argomenti? BASIC a vari livelli, programmi applicativi,



grafica, musica, Logo. La stimolante iniziativa, che rientra nelle attività di diffusione della cultura informatica svolte dal Commodore Computer Center Tempo Reale, potrà presto essere allargata ad altre città, grazie all'apertura di centri gemelli in diverse parti d'Italia.

Tempo Reale
Via Centotrecento, 1/A
40126 Bologna
Tel. 051-270701

J.*soft:* nuovo distributore dei prodotti Lotus

Windsor, UK-Marzo.

Lotus Development European Corporation, filiale europea della notissima software house statunitense, annuncia la nomina del proprio nuovo distributore italiano: J.*soft* di Milano.

"Man mano che anche in Italia si sviluppa un immenso potenziale di crescita del mercato del software standard per personal computer - commenta David Mc Donald, responsabile Lotus per la distribuzione in Europa - è assolutamente fondamentale che noi ci prepariamo a supportarlo in modo adeguato. La nomina di un distributore forte e professionalizzato è un elemento-chiave e noi riteniamo che J.*soft* offra in pieno l'altissimo standard che noi richiediamo a un distributore nazionale dei nostri prodotti. Siamo sicuri che la collaborazione con J.*soft* potrà tradursi in un ottimo livello di servizio e supporto sia per i dealer sia per gli utenti finali".

J.*soft* distribuirà, inizialmente, attraverso la propria rete di 11 agenti su tutto il territorio nazionale, i prodotti di punta per l'IBM PC e compatibili, come il Lotus 1-2-3, che esce in questi giorno con un manuale in Italiano, e Symphony, che entro brevissimo tempo sarà disponibile in versione totalmente nazionalizzata.

Entro la fine di Aprile a tali prodotti si aggiungerà Jazz, il nuovo integrato Lotus per Apple Macintosh.

J.*soft* installerà una hot-line telefonica specificatamente dedicata agli utenti Lotus (in aggiunta, quindi, a quella già esistente dedicata a tutti gli altri prodotti J.*soft*) e sarà in grado di offrire un programma completo di training e supporto a tutti i dealer autorizzati Lotus.

"Con l'aggiunta dei prodotti Lotus - commenta soddisfatto Giorgio Bozzini, consigliere delegato J.*soft* - riteniamo che il nostro portafoglio prodotti possa dirsi veramente all'altezza di un mercato che anche nel nostro Paese sta raggiungendo, a tempi stretti, dimensioni e professionalità al passo con gli altri paesi europei".

*J.*soft
V.le Restelli, 5
20124 Milano
Tel. 02-683797/6888228

Plotter low cost da San Marino

Un'azienda, della più antica repubblica del mondo, affronta, affiancata dall'esperienza giapponese, il mercato delle periferiche. Si tratta dell'ACS (Advance Computer Sistem) che presenta il plotter stampante PS/80, capace di 4 colori e

ottanta colonne per fogli singoli o rulli in formato massimo A4.
Questo plotter, assicurano all'ACS, è tra i più economici nella sua fascia di prestazioni. Il PS/80 può scrivere su qualunque tipo di carta o cartoncino disegni e caratteri variabili in altezza da 2 a 13 mm; con pennini speciali è possibile lasciare tracce anche su pellicola. Le istruzioni grafiche sono 13 e le interfacce previste sono la Centronics (di serie) o la opzionale RS-232. Assistenza e ricambi vengono garantiti direttamente dall'ACS.

Graf
Via Lagrangia, 6
47037 Rimini - Bellariva (FO)
Tel. 0541-33419

Tecnologie educative a convegno

Il nostro paese ha ospitato a Bologna i lavori della sessione 1985 del progetto ANTEM. Questo programma, la cui sigla sta per Applications des Nouvelles Technologies Educatives Multimedia, è uno dei 18 previsti all'interno del gruppo Technologie Croissance Enmploi (TCE), istituito al vertice di Versailles del 1982 tra i sette paesi più industrializzati. Francia e Canada partecipano ad ANTEM in qualità di paesi piloti, mentre Italia, Inghilterra e Comunità Europee sono associati a questo progetto che intende promuovere la collaborazione internazionale per lo sviluppo e la diffusione delle nuove tecnologie applicate alla didattica. L'iniziativa, che ha visto l'adesione di quattro ministeri, di ricercatori universitari internazionali e di numerosi esperti di informatica applicata alla didattica, si è svolta a Bologna il 30 e 31 Gennaio ed il primo Febbraio, ed è stata promossa da enti quali il CNR ITD, lo CSATA, Dioikema che ne ha curato la segreteria, l'ENEA, l'IRRSAE e l'Istituto di Cibernetica dell'Università di Milano.

Dioikema
Lorena Nanni
Via S. Stefano, 16
40125 Bologna
Tel. 051-556475

Othello e computer

Per gli appassionati di questo gioco riportiamo l'appello lanciato all'ultimo campionato mondiale tenutosi a Melbourne verso la fine del 1984. L'associazione giapponese dei giocatori di Othello ha in animo di pubblicare un volume che riporti tutte le notizie relative al gioco ed alle sue realizzazioni su microcomputer. Se avete sviluppato qualcosa che merita di essere segnalato, scrivete all'indirizzo sotto riportato, spiegando i criteri che vi hanno guidato nella realizzazione.

Kohko Kunieda
Suginami - Ku Ogikubo 3-45-8
Tokyo - Japan

Tascabili, che passione!

Dalla Francia segnaliamo una pubblicazione periodica rivolta ai patiti delle calcolatrici portatili programmabili, ed in particolare ai possessori dei modelli HP nelle diverse variazioni (41, 75, 71, 1X), nonché ai cultori del Forth. Si tratta di Micro Revue, un bimestrale curato dal club de Programmation Personelle sur Calculatrice de Toulouse (PPC-T) che presenta giochi, programmi, piccoli progetti hardware dedicati al mondo dell'informatica portatile. Poiché nel nostro paese si riscontra la mancanza quasi assoluta di letteratura in questo settore, riportiamo l'indirizzo del Club, segnalando che l'abbonamento costa 250 franchi francesi per 6 numeri inviati per posta aerea, mentre il singolo fascicolo costa 50 franchi.

Micro-Revue
PPC-T
77 Rue de Cagire
31100 Toulouse
France

Aula multimediale a Milano

Proseguendo nella rassegna delle iniziative che legano informatica e didattica, sempre più frequenti in questo periodo, segnaliamo la realizzazione di un'aula dall'impegnativa denominazione di "multimediale" realizzata presso la scuola media Carmelita Manara di Milano dalla cooperativa Il Dialogo. Questo laboratorio permanente, realizzato all'interno di una struttura scolastica esistente, dispone di nove computer Commodore, un videoproiettore da 45 pollici, un videoregistratore ed una lavagna luminosa con schermo. Completano la dotazione programmi didattici e videocassette relativi alle diverse materie trattate. Le funzioni dell'aula sono molteplici: è possibile usarla come videoteca didattica, per ospitare corsi di introduzione all'informatica, come laboratorio per la creazione di audiovisivi su videocassetta corredati di titoli elettronici e immagini generate da programmi appositi.

Il Dialogo S.c.r.l.
Viale Molise, 5
20137 Milano
Tel. 02-5460880



Routine in linguaggio macchina

Speciale per utenti ZX81

di *Carlo Cappelli*

Nel lontano numero 18 di **Personal Software**, ci siamo addentrati nell'affascinante mondo della programmazione in linguaggio macchina; vediamo ora come integrare linguaggio macchina e BASIC per una maggiore possibilità di utilizzo delle routine fino ad ora presentate. Per incominciare, vogliamo proporvi una interessante applicazione della routine per le animazioni, presentata nel numero già citato di **Personal Software**; lo scopo che ci siamo proposti, è quello di memorizzare i grafici di funzioni matematiche e poi alternarli velocemente sul video ottenendo animazioni incredibili. Per realizzare ciò bisogna innanzitutto caricare in memoria le routine precedentemente citate, costituite da una prima Rem con il programma per l'invio dei fotogrammi sullo schermo, e da una seconda Rem con il programma per gestire i vari fotogrammi.

Si devono apportare delle modifiche eseguendo delle Poke:

POKE 16633,198
POKE 16637,35
POKE 16638,35
POKE 16639,35

La necessità di queste modifiche è dovuta al fatto che non utilizziamo più una normale stringa A$ come in precedenza, bensì sfruttiamo una stringa A$ inizialmente dimensionata; questa operazione iniziale è necessaria per preparare quella parte di memoria nella quale inserire i vari grafici.

A causa di questo dimensionamento, il codice di A$ non è più 70 ma 198, e varia anche la posizione dei dati riguardanti la lunghezza della stringa stessa che si trova tre byte più avanti. (Si notino a questo proposito le tre istruzioni Inc Hl che vengono inserite dalla locazione 16637).

Fatta questa operazione, bisogna inserire il listato 1 che contiene un programma in BASIC per tracciare il grafico delle funzioni e alcune istruzioni per inizializzare le routine in linguaggio macchina. Le prime Poke servono per definire le dimensioni dell'immagine che deve essere contenuta nelle prime 21 linee e nelle prime 31 colonne, si tralasciano, quindi, l'ultima linea e l'ultima colonna dello schermo.

Al primo input bisogna inserire il numero di immagini che si desidera comporre; a questo proposito facciamo notare che con 16 Kbyte si possono inserire fino a 21 immagini (ogni immagine occupa 651 byte).

Al secondo e terzo input dobbiamo inserire il centro degli assi, in riferimento ai quali verrà tracciato il grafico della funzione.

Per posizionarlo, al centro dello schermo, bisogna assegnare il valore 30 alla X e 20 alla Y; se invece lo si vuol lasciare nell'angolo in basso, a sinistra, basta dare il valore 0 sia alla X che alla Y.

Il quarto input richiede l'inserimento di un coefficente che rappresenta il grado di ingrandimento dell'immagine; il valore da inserire va da uno a quattro; oltre non è consi-

Listato 1. *Programma per realizzare l'animazione di funzioni; traccia il grafico delle funzioni e le alterna velocemente sullo schermo.*

```
   1 REM   3+ E£RND7- U RND? 5;(
 CLEAR   RND?; FOR U RNDM RND5IN
KEY$?U RND? GOSUB   FAST STR$ ?Y
5-  LPRINT  ?; FOR LPRINT U RNDX
COS M RND/ STOP ..........
   5 REM E RND- ??5  ;( CLEAR
 FOR 5  ?;( CLEAR FOR M(INKEY$E(
 ND? RETURN ?C= 4 LET 76 RNDTAN
 (M*RNDTAN  LN  RND6 RNDU(INKEY$
 ASN PEEK RNDM(INKEY$TAN .......
 ................................
 ................................
 ..........
  10 REM ANIMAZIONI DI FUNZIONI
  20 POKE 16516,31
  30 POKE 16517,21
  40 POKE 16514,0
  50 POKE 16515,0
  60 PRINT "NUMERO IMMAGINI? ";
  70 INPUT N
  75 POKE 16609,N
  80 PRINT N
  83 LET A$=""
  84 GOTO 90
  85 FAST
  86 FOR P=1 TO N*600
  87 LET A$=A$+" "
  88 NEXT P
  89 SLOW
  90 LET P=1
 100 PRINT
 110 PRINT "INSERISCI IL CENTRO
DEGLI ASSI"
 120 PRINT "PRIMA LA X "
 130 INPUT CX
 140 PRINT CX
 150 PRINT "ORA LA Y ";
 160 INPUT CY
 170 PRINT CY
 180 PRINT "INSERISCI IL COEFFIC
```




```
ENTE DI IN  GRANDIMENTO DA 1 A 4
"
 190 INPUT G
 200 CLS
 210 PRINT "INSERISCI LA FUNZION
E IN X"
 220 INPUT X$
 230 CLS
 240 LET X1=-CX/G
 250 LET X2=(60-CX)/G
 270 FOR X=X1 TO X2 STEP .47/G
 280 LET Y=VAL X$
 290 LET I=X*G+CX
 300 LET Y=Y*G+CY
 310 IF Y>40 OR Y<0 THEN GOTO 33
0
 320 PLOT I,Y
 330 NEXT X
 340 LET DF=1+PEEK 16396+256*PEE
K 16397
 345 FAST
```

```
 350 FOR J=0 TO 20
 360 FOR K=0 TO 30
 370 LET A$=A$+CHR$ PEEK (DF+J*3
3+K)
 380 NEXT K
 390 NEXT J
 395 SLOW
 400 LET P=P+1
 410 IF P=N+1 THEN GOTO 460
 420 PRINT AT 20,0;"VUOI CAMBIAR
E I PARAMETRI"
 430 INPUT X$
 435 CLS
 440 IF X$="S" OR X$="SI" THEN G
OTO 100
 450 GOTO 210
 460 RAND USR 16595
 470 RAND USR 16657
 480 FOR U=1 TO 4
 490 NEXT U
 500 GOTO 470
```

gliabile. Se si inserisce ad esempio il valore uno, la funzione viene studiata in un intervallo di 60 unità, se si inserisce 3 essa viene studiata in un intervallo di 20 unità. In particolare, se al coefficiente di ingrandimento è stato dato il valore di 3 e il centro degli assi è posizionato in 30,20, la funzione verrà analizzata da —10 a 10 e i valori della Y potranno andare da —6.6 a 6.6. È opportuno rendersi conto di questi valori per poter posizionare nel migliore dei modi le funzioni scelte.
Le regole matematiche che individuano gli estremi entro i quali viene tracciata la funzione, sono le seguenti:

X minima = —CX/G
X massima = (60-CX)/G
Y minima = —CY/G
Y massima = (40-CY)/G.

CX e CY rappresentano le coordinate del centro, mentre G esprime il coeffi ciente di ingrandimento. Al quinto input bisogna inserire la funzione in X ricordando che si deve scrivere solo la parte contenente la X. Ad esempio la funzione Y uguale X*X deve essere inserita come X*X tralasciando /Y = ". Nel caso di funzioni che scendono rapidamente, per ridurre la discontinui tà del tracciato, conviene ridurre il valore dello Step della linea 270.

Vediamo ora cosa si può realizzare con questo programma ponendo il centro degli assi AN (30,20) e G = 1. Possiamo ad esempio rappresentare il movimento delle ali di un uccello inserendo varie parabole con aperture differenti; provate ad inserire queste 6 funzioni: X*X/40, X*X/72, -X*X /100, -X*X/64, -X*X/120, X*X/64. È meglio usa re X*X che non X↑2 (dove ↑ rappresenta l'elev amento a potenza), poiché, come su molti calcolatori, quest'ultima scrittura non accetta per la X valori negativi. Questo discorso vale per tutti gli elevamenti a potenza, non solo per i quadrati.
Se volete stilizzare il movimento di un'elica inserite queste 4 funzioni: Abs X *X/10, Abs X*X/64, -Sgn X* Sqr Abs(X* 4), -Sgn X* Sqr Abs (X*40). Potete provare a digitare 4 vo lte la funzione X* Sin X* inizialmente con G = 2 poi con G = 3, G = 4 e infine ancora con G = 3; per modificare il coefficiente G, dovrete aspettare fino a che il programma non vi chieda se volete cambiare i parametri; a questa domanda rispondete di sì, dopo di che, dovrete reinserire il centro de gli assi e dare il nuovo valore di G. Così facendo, otterrete un progressivo ing randimento e rimpicciolimento della funzione dall'effetto molto suggestivo. Il programma successivo serve per realizzare un linguaggio grafico simile per alcuni aspetti al Logo; con esso, possiamo comporre facilmente disegni, memorizzarli e richiamarli quando vogliamo.

Il principio di funzionamento è semplice; si deve stabilire il punto di partenza, da lì, con degli spostamenti in otto direzioni, si dovrà comporre l'immagine. Abbiamo a disposizione una tartaruga immaginaria che, muovendosi, lascia una traccia sullo schermo; il percorso che la nostra tartaruga descrive, corrisponde all'insieme di istruzioni che le abbiamo dato. Queste istruzioni devono essere contenute in una stringa così le possiamo salvare su cassetta e salvare con esse il disegno che rappresentano. A questo scopo utilizziamo la stringa A$ che non deve contenere



Routine in linguaggio macchina

Figura 1. *Istruzioni in linguaggio macchina con relativi codici decimali del programma per realizzare un linguaggio grafico.*

16514	—	0
	—	0
	—	0
	—	0
16518	POP HL	225
16519	LD A.20	62 20
	LD (16384).A	50 0 64
	RET	201
16525	LD HL. (16438)	42 54 64
	LD (L6514).HL	34 130 64
	LD A.0	62 0
	LD (16516).A	50 132 64
	LD HL. (16400)	42 16 64
	LD A.H	124
	CP 125	254 125
	JR Z —24	40 231
	LD A.(HL)	126
	CP 70	254 70
	INC HL	35
	JR NZ —10	32 245
	LD A.(HL)	126
16551	CP O	254 0
	JR NZ —10	32 245
	INC HL	35
	INC HL	35
16557	LD A. (16552)	58 168 64
	LD B.A	71
	LD A.(HL)	126
16562	CP 38	254 38
	JR C —46	56 209
	LD C.128	14 128
	CP 127	254 127
	JR C 4	56 4
	LD C.64	14 64
	SUB 128	214 128
	LD E.1	30 1
	LD (16517).A	50 133 64
	LD A.C	121
	LD (16803).A	50 163 65
	INC HL	35
	DEC B	5
	JR Z 40	40 40
	LD A.(HL)	126
	CP 38	254 38
	JR NC 35	48 35
	CP 29	254 29
	JP C 16519	218 135 64
	SUB 28	214 28
	LD C.A	79
	INC HL	35
	DEC C	5
	JR Z 22	40 22
	LD A.(HL)	126
	CP 38	254 38
	JR NC 17	48 17
	SUB 28	214 28
	JP C 16519	218 135 64
	LD D.A	87
	LD A.C	121
	ADD A.A	135
	ADD A.A	135
	ADD A.A	135
	ADD A.C	129
	ADD A.C	129
	ADD A.D	130
	LD E.A	95
	INC HL	35
	JR 2	24 2
	LD E.C	89
	INC B	4
	PUSH HL	229
	LD A.(16517)	58 133 64
	CP 40	254 40
	JR NZ 7	32 7
	LD HL.16516	33 132 64
	LD (HL).1	51 1
	JR 107	24 107
	CP 53	254 53
	JR NZ 7	32 7
	LD HL.16516	33 132 64
	LD (HL).0	54 0
	JR 96	24 96
	LD HL.0	33 0 0
	CP 38	254 38
	JR NZ 2	32 2
	LD L.36	46 36
	CP 39	254 39
	JR NZ 2	32 2
	LD L.37	46 37
	CP 41	254 41
	JR NZ 3	32 3
	LD HL.11264	33 0 44
	CP 56	254 56
	JR NZ 3	32 3
	LD HL.11520	33 0 45
	CP 54	254 54
	JR NZ 3	32 3
	LD HL.11300	33 36 44
	CP 60	254 60
	JR NZ 3	32 3
	LD HL.9516	33 44 37
	CP 42	254 42
	JR NZ 3	32 3
	LD HL.11557	33 37 45
	CP 55	254 55
	JR NZ 3	32 3
	LD HL.9261	33 45 36
	LD (16720).HL	34 80 65
	LD C.E	75
	LD HL.(16514)	42 130 64
16720	NOP	0
	NOP	0
	LD A.H	124
	CP 44	254 44
	JP NC 16518	210 134 64
	LD A.L	125
	CP 64	254 64
	JP NC 16518	210 134 64
	LD A.(16516)	58 132 64
	LD (16514).HL	34 130 64
	CP 1	254 1
	JR Z 5	40 5
	PUSH BC	197
	CALL 16804	205 164 65
	POP BC	193
	DEC C	13
	JR NZ —34	32 221
	POP HL	225
	DEC B	5
	JP NZ 16561	194 177 64
16757	RET	201

più di 255 caratteri; tale valore corrisponde a circa 150 istruzioni. A questo punto analizziamo le istruzioni a disposizione.

A - La tartaruga si sposta verso l'alto rispetto al punto in cui si trovava prima; la lettera può essere seguita da un numero di una o due cifre che rappresentano il numero di caselle che la tartaruga deve percorrere; nel caso mancasse il numero, si sposta di una sola casella.

B - Come prima, però lo spostamento avviene verso il basso.

D - Come prima, però lo spostamento avviene verso destra.

S - Come prima, però lo spostamento avviene verso sinistra.

Q - Come prima, però lo spostamento avviene verso nord-est.

W - Come prima, però lo spostamento avviene verso sud-est.

E - Come prima, però lo spostamento avviene verso sud-ovest.

R - Come prima, però lo spostamento avviene verso nord-ovest.

Possiamo ad esempio scrivere A$ = "A3BDDB15R7", con questa sequenza di istruzioni la nostra tartaruga andrà in alto di tre caselle poi in basso di una, a destra di una, a destra di una ancora, in basso di 15 e infine a nord-ovest di 7. I valori da dare agli spostamenti devono essere tali da non fare uscire dallo schermo la tartaruga, altrimenti il programma si fermerà con errore 1. Le lettere A,B,D,... possono essere scritte anche in campo inverso, in tal caso, la tartaruga lascerà una traccia bianca da abbinarsi con lo sfondo annerito. Abbiamo ancora due istruzioni da chiarire: la prima è C, con questo comando la tartaruga non lascia alcuna traccia dei suoi spostamenti; mentre con il comando P la nostra bestiola ricomincia a lasciare la sua traccia. Passiamo ora a descrivere la realizzazione del programma in linguaggio macchina; per far ciò, occorre caricare in memoria il programma del Draw veloce (quello con 1 Rem...e 5 Rem...) presentato nella prima parte di questa rassegna di routine in Assembly.

In questo programma sfrutteremo interamente la 5 Rem mentre modificheremo sostanzialmente la 1 Rem.

Per inserire i codici usufruiamo della solita routine di caricamento con 10 Let A = 16514; i codici da introdurre sono quelli di figura 1.

La prima parte della routine serve a rintracciare la stringa A$; dalla locazione 16557 si incomincia a controllare i caratteri della sequenza di istruzioni, distinguendo i numeri preceduti dalle lettere chiave. Fatto questo, il programma assegna a ogni lettera l'insieme di istruzioni ad essa legate; queste istruzioni, in linguaggio macchina, vengono poste nelle locazioni 16720 e 16721, successivamente il programma le esegue un numero di volte pari a quello delle caselle da percorrere. La stampa sullo schermo avviene attraverso una chiamata alla routine di plottaggio di 5 Rem.

Per rendere operativo il programma, è necessario inserire la lunghezza della stringa A$ con Poke 16552, Len A$; poi eseguire la routine con Rand USR 16525.

Per rendervi conto del funzionamento, provate ad inserire le istruzioni BASIC del listato 2. Per sfruttare al meglio questo linguaggio grafico provate i tre programmi del listato 3 (per attivare il secondo fare Goto 100, mentre per il terzo Goto 150).

Le possibilità applicative sono molte, basta sbizzarrire la fantasia e anche con un piccolo ZX81 si possono fare cose grandi (vedi figura 2).

Prendiamo ora in considerazione un altro argomento; vediamo come implementare alcuni principi di intelligenza artificiale nella programmazione in Assembly. A questo scopo vediamo di realizzare il classico gioco del Master Mind nel quale il calcolatore sia in grado, sia di codificare che di decodificare. Programmi di questo tipo se ne sono già visti parecchi; ma questo differisce da tutti gli altri (oltre al fatto di essere scritto in linguaggio macchina), per la sua grande adattabilità a svariate situazioni; ad esempio, può essere utilizzato per inserire dei tentativi ed analizzare le possibili soluzioni. Il pregio principale consiste nella possibilità di personalizzare le tecniche di ricerca del codice da indovinare, con

Listato 2. *Esempio d'utilizzo della routine di figura 1.*

```
   1 REM     LPRINT Y=M RNDTAN E
QRND6▙RNDY M▜RNDE(RND? RETURN ?C
 SCROLL 7U█RND?█ RETURN █5▖:RNDC
HR$ █2▘M ▌RND?M█INKEY$7▌CC1 AND ▖
RNDCHR$ 077▌C-0 AND ▖RND??▖▖▟▀▙
?7/▝?▖ FAST U▐RND RETURN C4▛5▜RN
DQ▘/? RETURN P4▛5▜RNDQ /?5  RET
URN A4▝I8 RETURN B4▝I9 RETURN D4
▀5 G RETURN S4▀5 H RETURN Q4▀58G
 RETURN W4▀5G9 RETURN E4▀59H RET
URN R4▀5H86?INKEY$?E▙RND8 ? RETU
RN GABS ▚RND? RETURN RNDABS ▚RND
U▜RND6▙RND RETURN ▘C▌VAL LN █INK
EY$AT $4<> LPRINT ▌TAB █RNDTAN .
::::::::::::::::::::::::::::::::

   5 REM █?█  YF█?:  RETURN █5▖£
£CHR$ █M█INKEY$?? RETURN █S▀£CHR
$ █- ?E£RND;U█INKEY$?7▖▝5;( CLEAR
 6█INKEY$▖▝▒??52PI;( CLEAR ?▖▀▒U█I
NKEY$ RETURN RNDC57TAN RETURN █5
▀█/▘▄E█INKEY$?? RETURN ▖""█?TAN
7( RUN TAN RETURN █6▀▖/▘█/<=▝ ▖
:::::::::::::::::::::
  10 FOR I=1 TO 20
  20 PRINT "████████████████"
  30 NEXT I
  40 UNPLOT 2,3
  50 LET A$="█4█5█6CR7A3P█12███D
17A6Q8D6QB5W3PD4A3"
  60 POKE 16552,LEN A$
  70 LET K=USR 16525
```

Routine in linguaggio macchina

Listato 3. *Con questo programma è stata ottenuta la figura 2.*

```
   1 REM ;■ A LPRINT Y=M RNDTAN
EQRND8■RNDY M■RNDE(RND? RETURN ?
C SCROLL 7U■RND7■ RETURN ■S■:RND
CHR$ ■2" M ■RND?M■INKEY$7■CC1 AND
■RNDCHR$ 077■C-0 AND ■RND?? ■■■
■77/ "?■ FAST U■RND RETURN C4■5■R
ND0" /? RETURN P4■5■RNDQ /?5   RE
TURN A4 "I8 RETURN B4 "I9 RETURN D
4■5 G RETURN S4■5 H RETURN Q4■58
G RETURN W4■5G9 RETURN E4■59H RE
TURN R4■5H86?INKEY$?E■RND8 ? RET
URN GABS ■RND? RETURN RNDABS ■RN
DU■RND8■RND RETURN "C■VAL LN ■IN
KEY$AT $4<> LPRINT ■TAB ■RNDTAN
: : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : :
: : : : :
   5 REM ■?■· YF■?:  RETURN ■S■£
£CHR$ ■M■INKEY$?? RETURN ■S■£CHR
$ ■- ?E£RND;U■INKEY$?7■5;( CLEAR
 S■INKEY$■■??52PI;( CLEAR ?■■U■I
NKEY$ RETURN RNDC57TAN RETURN ■S
■■/"■E■INKEY$?? RETURN ■""■?TAN
7( RUN TAN RETURN ■S■■/■■/<= ■ ■
■ ■■■ ■ ■■ ■■■ ■■■■■ ■ ■■ ■■
■: : : : : : : : : : : : : : : : : : : : : : : :
  10 UNPLOT 5,36
  20 LET A$="BCBPBDA3B6DA6DB8CB1
0PB2CB3PBDB2A7CA10PA11DB9CB10PB9
0A8CA11PA7WB5CB11PB5EWB3CDA5PAAC
A12PA10WB8WA100B12CB3PBCA2PDA130
AB16DA16DB17WA19WB6CB3PB10DA7CA6
PA5DB4CB6PB9WA8CA7PA4BWBBCB7PB7W
A5WB5WA5WB3WBA5DB6DA4DB6WA7DBBCB
BPB7RB33A04AEAACAAPA4WBD3BBRAAS"
  30 POKE 16552,LEN A$
  40 LET K=USR 16525
  50 UNPLOT 33,5
  60 LET A$="BBEA6EB5RA35BB5AA5B
BAARBBRA"
  70 POKE 16552,LEN A$
  80 LET K=USR 16525
  90 STOP
 100 UNPLOT 0,0
 110 LET A$="D62A42S60B40D58A383
56B36D54A34S52B32D50A30548B28D46
A26S44B24D42A22S40B20D38A18S36B1
6D34A14S32B12D30A10S28B8D26A6S24
B4D22A2S20"
 120 POKE 16552,LEN A$
 130 LET K=USR 16525
 140 STOP
 150 DIM L$(11,19)
 160 LET L$(1)="DAA2S5A2D5CDB4CC
CCC"
 170 LET L$(2)="DA4CAPACB6DCCCCC
CCC"
 180 LET L$(3)="DA4D5B4CDCCCCCCC
CCC"
 190 LET L$(4)="D6S5A4D5CB4DCCCC
CCC"
 200 LET L$(5)="DA6B6CDCCCCCCCCC
CCC"
 210 LET L$(6)="D6A4S5CBPBBQD4CB
BDC"
 220 LET L$(7)="DA4D5CB4DCCCCCCC
CCC"
 230 LET L$(8)="DQ5S5CB5PD5CDCCC
CCC"
 240 LET L$(9)="DQ5CS4PSW5CDCCCC
CCC"
 250 LET L$(10)="QA0RQDDWESDWBES
5CD4"
 255 LET L$(11)="DA5CB5DCCCCCCCC
CCCC"
 260 UNPLOT 0,20
 270 POKE 16552,19
 280 LET A$=L$(1)
 285 LET K=USR 165?5
 290 LET A$=L$(2)
 300 GOSUB 600
 310 LET A$=L$(3)
 320 GOSUB 600
 330 LET A$=L$(4)
 340 GOSUB 600
 350 LET A$=L$(5)
 360 GOSUB 600
 370 LET A$=L$(6)
 380 GOSUB 600
 390 LET A$=L$(2)
 400 GOSUB 600
 410 LET A$=L$(7)
 420 GOSUB 600
 430 UNPLOT 34,10
 440 LET A$=L$(8)
 450 LET K=USR 16525
 460 LET A$=L$(9)
 470 GOSUB 600
 480 LET A$=L$(10)
 490 GOSUB 600
 500 LET A$=L$(11)
 510 GOSUB 600
 520 PRINT AT 20,0 "PREMI N/L"
 530 INPUT K$
 540 CLS
 550 GOTO 700
 600 LET K=USR 16531
 610 RETURN
 700 FOR I=1 TO 100
 710 UNPLOT RND*57 RND*37
 720 LET A$=L$(INT (RND*11+1))
 730 LET K=USR 16525
 740 NEXT I
```

semplici istruzioni BASIC. La descrizione del programma chiarisce quanto detto. Il programma è diviso in tre parti.

La prima serve a generare le 1.296 soluzioni possibili nella memoria del calcolatore; il numero dato equivale alle disposizioni con ripetizione di 6 elementi (corrispondenti ai sei colori del Master Mind) accoppiati a 4 a4. Tutto questo avviene nelle locazioni che vanno da 16530 a 16565.

La seconda parte del programma serve per valutare un tentativo rispetto alla combinazione posta nelle locazioni 16524-5-6-7. Questa parte inizia dalla locazione 16566 e termina a 16628.

La terza parte è quella più interessante; infatti qui, dato un tentativo e la sua valutazione, si analizzano tutte quelle combinazioni che a quel tentativo associano quella valutazione; tutte le altre combinazioni vengono eliminate. Per ottenere questo risultato si pone la possibile soluzione nelle locazioni 16524-5-6-7, quindi si valuta rispetto ad essa il tentativo dato; se il responso coincide con la valutazione fornita, allora quella soluzione risulta ancora possibile, altrimenti viene eliminata. Procedendo con un altro tentativo si ridurrà ulteriormente il numero delle soluzioni possibili finché ne rimarrà una sola; nel caso si commettano

Routine in linguaggio macchina

degli errori nelle valutazioni dei tentativi, il programma, non trovando più soluzioni possibili, si fermerà dando errore H. Per inserire questa routine occorre, come al solito, il programma di caricamento con Let A = 16514 e una Rem di almeno 250 caratteri.
Fatto questo, basta inserire i codici della figura 3.

A questo punto bisogna realizzare un programma BASIC che sfrutti le routine in linguaggio macchina. Per fare in modo che siate voi a decodificare la combinazione, dovete generare casualmente una sequenza di 4 cifre dall'uno al sei nelle locazioni 16524-5-6-7 (vedi linee 120, 130, 140, 150 del listato 4); i tentativi per indovinare quella combinazione devono essere inseriti nelle locazioni 16520-1-2-3; per la valutazione si ricorre alla seconda routine che viene chiamata con Let K = Usr 16629 (si noti l'uso di Let.. invece di Rand;... per i motivi illustrati nell'articolo precedente) il responso si troverà nelle locazioni 16528-9; nel primo byte vi sarà il numero di elementi presenti al posto giusto, mentre nel secondo byte gli elementi presenti nel posto sbagliato.
Se volete invece che sia il computer a scoprire il vostro codice, dovete generare tutte le soluzioni possibili con una chiamata alla prima subroutine tramite Let K = Usr 16530; il primo tentativo del calcolatore deve essere generato casualmente e posto nelle locazioni a partire da 16524, mentre la valutazione del tentativo deve essere collocata nei byte 16514 e



Figura 2. *Il "Bel Paese" come viene mostrato attraverso il linguaggio grafico implementato con la routine della figura 1.*

Figura 3. *Istruzioni in linguaggio macchina per il Master Mind.*

16514	—	0
	—	0
	—	0
	—	0
	—	0
	—	0
	—	0
	—	0
	—	0
	—	0
	—	0
	—	0
	—	0
	—	0
	—	0
16529	—	0
16530	LD HL.25000	33 168 97
	LD B.6	6 6
	LD E.6	30 6
	LD D.6	22 6
	LD C.6	14 6
	LD (HL).B	112
	INC HL	35
	LD (HL).E	115
	INC HL	35
	LD (HL).D	114
	INC HL	35
	LD (HL).C	113
	INC HL	35
	DEC C	13
	JR NZ -10	32 245
	DEC D	21
	JR NZ -15	32 240
	DEC E	29
	JR NZ -20	32 235
	DJNZ -24	16 231
	LD (HL).117	54 117
	LD (16760).HL	34 120 65
16565	RET	201
16566	LD HL.16516	33 132 64
	LD DE.16520	17 136 64
	LD C.0	14 0
	PUSH HL	229
	LD B.4	6 4
	LD A. (DE)	26
	CP (HL)	190
	JR NZ 7	32 7
	INC C	12
	LD (HL).0	54 0
	EX HL.DE	235
	LD (HL).8	54 8
	EX HL.DE	235
	INC HL	35
	INC DE	19
	DJNZ -14	16 241
	LD A.C	121
	LD (16528).A	50 144 64
	POP HL	225
	LD C.0	14 0
	LD B.4	6 4
	LD A.(HL)	126
	PUSH HL	229
	LD E.4	30 4
	LD HL.16520	33 136 64
	CP (HL)	190
	JR NZ 5	32 5
	INC C	12
	LD (HL).8	54 8
	JR 4	24 4
	INC HL	35
	DEC E	29



Routine in linguaggio macchina

Seguito figura 3.

	JR NZ -11	32 244
	POP HL	225
	INC HL	35
	DJNZ -22	16 233
	LD A.C	121
	LD (16529).A	50 145 64
16628	RET	201
16629	LD HL.16524	33 140 64
	LD DE.16516	17 132 64
	LD BC.4	1 4 0
	LDIR	237 176
	CALL 16566	205 182 64
16643	RET	201
16644	LD HL.25000	33 168 97
	LD A.(HL)	126
	CP 117	254 117
	JR NZ 6	32 6
	LD A.16	62 16
	LD (16384).A	50 0 64
	RET	201
	PUSH HL	229
	LD BC.4	1 4 0
	LD HL.16524	33 140 64
	LD DE.16516	17 132 64
	LDIR	237 176
	POP HL	225
	LD DE.16520	17 136 64
	LD BC.4	1 4 0
	LDIR	237 176
	PUSH HL	229
	CALL 16566	205 182 64
	POP HL	225
	LD A.(16514)	58 130 64
	LD C.A	79
	LD A.(16528)	58 144 64
	CP C	185
	JR Z 21	40 21
	LD BC.4	1 4 0
	DEC HL	43
	EX HL.DE	235
	LD HL.(16760)	42 120 65
	DEC HL	43
	LDDR	237 184
	INC HL	35
	INC DE	19
	LD (HL).117	54 117
	LD (16760).HL	34 120 65
	EX HL.DE	235
	JR 10	24 10
	LD A.(16515)	58 131 64
	LD C.A	79
	LD A.(16529)	58 145 64
	CP C	185
	JR NZ -30	32 225
	LD A.(HL)	126
	CP 117	254 117
	JR NZ -71	32 184
	LD DE.25000	17 168 97
	SBC HL.DE	237 82
	LD A.(DE)	26
	CP 117	254 117
	JP Z 16652	202 12 65
	SRL L	203 61
	SRL L	203 61
	SRL H	203 60
	JR NC 2	48 2
	SET 6.L	203 245
	SRL H	203 60
	JR NC 2	48 2
	SET 7.L	203 253
	LD B.H	68
	LD C.L	77
16759	RET	201
16760	—	0
	—	0

16515. Affinché il calcolatore analizzi tali valori è necessaria una chiamata alla terza subroutine tramite Let K = Usr 16644; dopo questa chiamata la variabile K assumerà un valore pari al numero di soluzioni possibili. È interessante vedere come questo numero si abbassi rapidamente man mano si inseriscono le valutazioni relative ai tentativi del computer; si può ad esempio notare che il numero delle possibili soluzioni, dopo il primo tentativo, passa da 1.296 a circa 200.

In base a questa analisi si potrà sviluppare il miglior metodo secondo cui scegliere i tentativi da giocare. È questo metodo che renderà il calcolatore più o meno "intelligente" infatti, il problema principale consiste nel come il calcolatore debba scegliere i tentativi successivi.

Questa scelta può essere completamente casuale, basta operare come per il primo tentativo; tuttavia, questo metodo porta alla soluzione dopo 7 o 8 tentativi. Una tecnica più efficace consiste nell'estrarre i tentativi successivi dall'insieme delle soluzioni possibili che si trovano a partire dalla locazione 25000; ogni tentativo dovrà poi essere inserito nelle locazioni 16524-5-6-7 e si dovrà operare come per il primo. In questo caso si presentano varie alternative; infatti, il nostro tentativo può essere estratto all'inizio o alla fine o in mezzo all'insieme delle soluzioni possibili; questo comporta una maggiore o minor velocità nel decodificare la combinazione pensata dal giocatore. Nel programma del listato 4, ad esempio, il tentativo viene estratto all'inizio (vedi linee 410,420,430,440,450,460,470). Se volete che il tentativo venga estratto in mezzo e casualment e potete inserire: 415 Let T = Int (Rnd * K + 1) *4 e modi ficare la linea 420 come segue: 420 For I = 24996 + T To 24996 + T + 3. Si possono fare anche controlli sui tentativi estratti, ad esempio, scartando quelli con delle cifre che si ripetono, poiché danno poche informazioni sulla combinazione da scoprire. A questo punto, spetta a voi scegliere il miglior metodo secondo cui far giocare il calcolatore e in questo modo apprenderete, con l'analisi empirica, elementi di intelligenza artificiale. Il programma del listato 4 esprime le potenzialità delle routine in linguaggio macchina che, combinate in modi opport uni, possono dar luogo a situazioni interessanti coma ad esempio il secondo livello di gioco, nel caso in cui sia il giocatore a risolvere la combinazione del computer.

I QUADERNI JACKSON
di Personal Computer

Gianni Giaccaglini

Vivere
col Personal Computer

comprenderlo, adoperarlo,
subirne il fascino e le nevrosi

**In edicola,
a sole
lire 6.000.**

L'INFORMATICA TASCABILE.

Arrivano i Quaderni Jackson, tanti volumi monografici per conoscere bene il personal computer e l'informatica.

Nei quaderni Jackson c'è tutto quello che è importante sapere sui computer, la programmazione, i linguaggi, il software, le applicazioni e i nuovi sviluppi dell'informatica.

Quaderni Jackson: l'informatica a tutti i livelli, in una collana aperta, pratica, essenziale, aggiornata.

L'informatica tascabile per chi vuole saperne di più e compiere così un salto di qualità nel mondo di oggi e di domani.

Ogni mese, 2 volumi.

Volumi già pubblicati:

Gianni Giaccaglini **"Vivere col Personal Computer"**
Paolo Bozzola **"Dentro e fuori la scatola"**
Enrico Odetti **"Ed è subito BASIC Vol. I"**
*Paolo Capobussi
e Marco Giacobazzi* **"A ciascuno il suo Personal"**



**GRUPPO
EDITORIALE
JACKSON**

SAN FRANCISCO-LONDRA-MILANO

Listato 4. *Sfruttando la routine di figura-3 si può implementare il Master Mind.*

```
   1 REM                        5▓?▞▚2▞
-▞:▞?????? ?$4 PRINT +4 LIST 14
FOR ( SCROLL Q?6?INKEY$TAN 5▜RND
)▒RND   FAST ▞▖,▓4▛£0  FOR Q▒ FO
R ?<( LET ?M▓RND LPRINT : ▞▖RND▓
4▓£0▒/▖?14 POKE LPRINT ?( DIM ?M
▓RNDTAN 5▓RND)▜RND▘▖  GOSUB ▓LN
▓RNDTAN 5▓?(M RNDTAN FAST ▘▖ 5▓R
ND)▜RND GOSUB ▓ LPRINT )▒RND▘▖
GOSUB ▓ FAST LN ▓RND LPRINT U▙RN
D?U▓RND▓C+▘▖ F FOR E?INKEY$F GOS
UB ▓?<Q?6?INKEY$ FOR /▀U▄RND?U▓R
ND▓4 LPRINT ▓? GOSUB ?, RETURN ?
ASN £INKEY$ACS XACS XACS UK▘ACS
PRINT ACS UK▘ACS CLEAR ??TAN    .
..................
  10 PRINT "       ** MASTER MIND
**"
  20 PRINT
  30 PRINT "1  PER DECODIFICARE
LA COMBINAZIONE DEL COMPUTER"
  35 PRINT
  40 PRINT "2 SE VUOI CHE SIA IL
 COMPUER A   DECODIFICARE LA TUA
 COMBINAZIONE"
  45 PRINT
  50 PRINT "3 SE VUOI INSERIRE D
EI TENTATIVI ED ANALIZZARE LE PO
SSIBILI SOLUZIONI"
  60 INPUT A
  65 CLS
  70 IF A=1 THEN GOTO 100
  80 IF A=2 THEN GOTO 300
  90 IF A=3 THEN GOTO 680
  95 GOTO 10
 100 PRINT "INSERISCI IL LIVELLO
 1 O 2"
 105 INPUT A
 110 IF A=2 THEN GOTO 1000
 115 CLS
 120 POKE 16524,INT (RND*6+1)
 130 POKE 16525,INT (RND*6+1)
 140 POKE 16526,INT (RND*6+1)
 150 POKE 16527,INT (RND*6+1)
 160 PRINT ,,"INSERISI I TENTATI
VI"
 170 INPUT A$
 180 CLS
 190 LET N=1
 200 PRINT "TENT.";N;" ";A$,
 220 FOR I=1 TO 4
 230 POKE 16519+I,VAL A$(I)
 240 NEXT I
 250 RAND USR 16629
 260 PRINT "PG.";PEEK 16528;"  P
S.";PEEK 16529
 265 IF PEEK 16528=4 THEN PRINT
"HAI VINTO IN ";N;" TENTATIVI"
 270 IF PEEK 16528=4 THEN GOTO 2
000
 280 INPUT A$
 290 LET N=N+1
 295 GOTO 200
 300 LET K=USR 16530
 304 CLS
 305 LET N=2
 306 PRINT "DEVI VALUTARE I MIEI
 TENTATIVI"
 307 PRINT "SE VUOI VEDERE LE SO
LUZIONI POS SIBILI BATTI S"
 308 PRINT 1;"  ";
 310 FOR I=1 TO 4
 320 LET J=INT (RND*6+1)
 330 PRINT J;
 340 POKE 16523+I,J
 350 NEXT I
 360 INPUT A$
 365 IF A$="S" THEN GOTO 600
 370 POKE 16514,VAL A$(1)
 380 POKE 16515,VAL A$(2)
 390 PRINT ,A$
 395 IF A$(1)="4" THEN PRINT "HO
VINTO IN ";N-1;" MOSSE"
 396 IF A$(1)="4" THEN GOTO 2000
 400 LET K=USR 16644
 405 PRINT "POSSIBILITA▘=";K
 407 PRINT N;" ";
 410 LET C=16524
 420 FOR I=25000 TO 25003
 430 LET J=PEEK I
 440 PRINT J;
 450 POKE C,J
 460 LET C=C+1
 470 NEXT I
 580 LET N=N+1
 590 GOTO 360
 600 CLS
 605 PRINT "SE VUOI PROSEGUIRE P
REMI R"
 610 FOR I=25000 TO 24999+K*4 ST
EP 4
 620 PRINT PEEK I;PEEK (I+1);PEE
K (I+2);PEEK (I+3)
 625 IF INKEY$="R" THEN GOTO 640
 630 NEXT I
 632 IF INKEY$="" THEN GOTO 632
 640 CLS
 650 PRINT "VALUTA QUESTO TENTAT
IVO"
 660 PRINT N;" ";PEEK 16524;PEEK
 16525;PEEK 16526;PEEK 16527
 670 GOTO 360
 680 LET K=USR 16530
 690 PRINT "PER VEDERLE BATTI S
AL POSTO DEL TENTATIVO"
 705 PRINT "INSERISCI IL TENTATI
VO"
 710 INPUT A$
 715 PRINT A$
 718 IF A$="S" THEN GOTO 900
 720 FOR I=1 TO 4
 730 POKE 16523+I,VAL A$(I)
 740 NEXT I
 750 PRINT "INSERISCI LA VALUTAZ
IONE"
 760 INPUT A$
 770 PRINT A$
 780 POKE 16514,VAL A$(1)
 790 POKE 16515,VAL A$(2)
 800 LET K=USR 16644
 810 PRINT "SOLUZIONI POSSIBILI=
";K
 820 IF K=1 THEN PRINT "LA SOLUZ
IONE FINALE E▘ "
 830 IF K=1 THEN PRINT PEEK 2500
0;PEEK 25001;PEEK 25002;PEEK 250
03
 840 IF K=1 THEN GOTO 2000
 880 GOTO 700
 900 CLS
 905 PRINT "PER RITORNARE PREMI
R"
 910 FOR I=25000 TO 24999+K*4 ST
EP 4
 920 PRINT PEEK I;PEEK (I+1);PEE
K (I+2);PEEK (I+3)
```

Seguito figura 4.

```
 930 IF INKEY$="R" THEN GOTO 950
 940 NEXT I
 941 PRINT "PER PROSEGUIRE PREMI
N/L"
 942 INPUT A$
 950 CLS
 960 GOTO 700
1000 CLS
1005 LET K=USR 16530
1010 LET N=1
1020 PRINT "INSERISCI I TENTATIV
I"
1030 PRINT "TENT.";N;" ";
1035 INPUT A$
1040 PRINT A$,
1050 IF N>3 THEN GOTO 220
1055 FOR I=1 TO 4
1060 POKE 16523+I,VAL A$(I)
1070 NEXT I
1080 DIM C$(3,2)
1081 LET C$(1)="01"
1082 LET C$(2)="02"
1083 LET C$(3)="10"
1090 LET J=INT (RND*3+1)
1100 PRINT "PG.";C$(J,1);"  PS."
;C$(J,2)
1110 POKE 16514,VAL C$(J,1)
1120 POKE 16515,VAL C$(J,2)
1130 LET K=USR 16644
1140 IF N=3 THEN GOTO 1160
1145 LET N=N+1
1150 GOTO 1030
1160 LET C=16524
1170 FOR I=25000 TO 25003
1180 POKE C,PEEK I
1185 LET C=C+1
1190 NEXT I
1200 LET N=N+1
1210 GOTO 1030
2000 PRINT "SE VUOI RICOMINCIARE
 BATTI R"
2100 INPUT A$
2120 IF A$<>"R" THEN STOP
2130 CLS
2140 RUN
```

Qui il calcolatore, in un certo senso, bara poiché per i primi tre tentativi vi da responsi casuali di scarso valore (ad esempio un elemento al posto sbagliato) successiavamente, in base a quei risultati, il calcolatore sceglie la sua combinazione rendendovi quindi più difficile la strada per giungere alla soluzione. Questa situazione la si attua sfruttando adeguatamente le tre subroutine .

Il programma successivo soddisferà sicuramente coloro che utilizzano lo ZX81 a scopi gestionali o comunque devono trattare un gran numero di dati. Come tutti sanno le ricerche di dati in BASIC sono particolarmente lente, tali da rendere più comodo e veloce consultare elenchi scritti. Il programma di gestione dati che vi presentiamo ha delle caratteristiche notevoli; infatti, permette non solo la ricerca di interi nominativi, ma anche la ricerca di gruppi di lettere posti in qualsiasi posizione o in posizioni prefissate; inoltre, si possono effettuare ricerche successive sui dati trovati.

I dati devono essere inseriti in una matrice stringa A$ a due dimensioni (esempio A$ (17,24) è una matrice stringa composta da 17 righe, ognuna delle quali contiene al massimo 24 elementi). Il numero delle righe non deve superare 255, lo stesso vale per il numero degli elementi di ogni riga. Non è necessario inserire i dati in ordine alfabetico o secondo altre sequenze; infatti, il programma non utilizza tecniche di ricerca particolari come ad esempio nella ricerca binaria o nella ricerca hashing, ma applica una ricerca sequenziale, cioè analizza tutta la lista di dati fino a incontrare l'elemento cercato. Normalmente questo tipo di ricerca non è molto apprezzato a causa dei lunghi tempi di elaborazione; ma questo discorso non vale se la programmazione avviene in linguaggio macchina; in questo caso il tempo perso nella ricerca si compensa con la semplicità e velocità dell'algoritmo usato.

Il dato, del quale vogliamo trovare la posizione e le ulteriori informazioni ad esso connesse, deve essere inserito nella stringa W$; il primo carattere di questa stringa deve esprimere la posizione del nostro dato all'interno di una riga della matrice A$. Se la posizione non è nota, il primo elemento di W$ deve essere posto a zero.

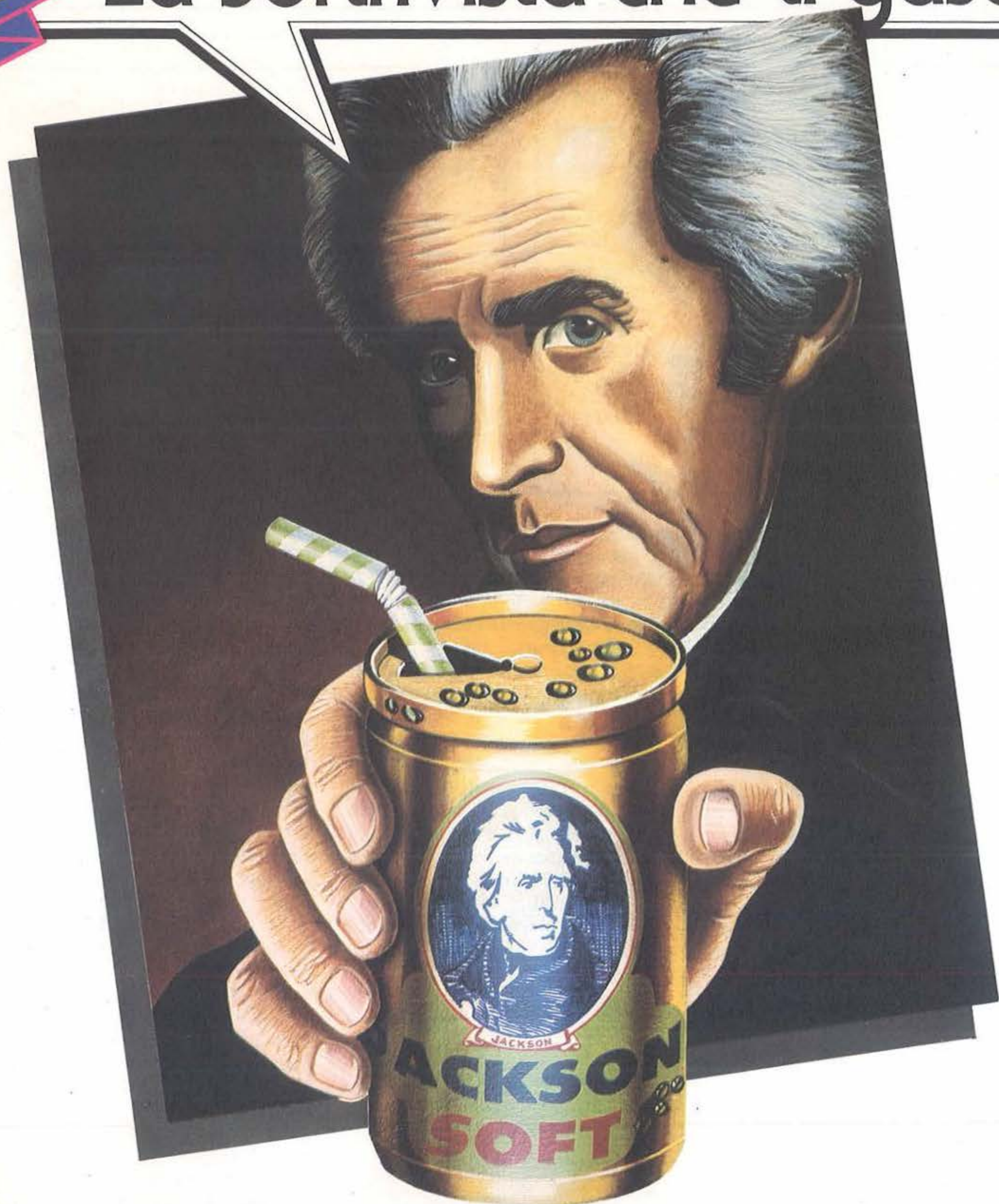
Per illustrare meglio quanto detto possiamo fare il seguente esempio: supponiamo di utilizzare il programma come un'agenda telefonica; in una riga della matrice stringa A$ memorizzeremo il cognome di un nostro amico, poi l'indirizzo e il numero di telefono; per sapere il recapito di questo amico è necessario conoscere il numero della riga in cui sono memorizzati i suoi dati. Per ottenere questo risultato, dovremo inserire nella stringa W$ il suo cognome preceduto da Chr$ 1 poiché il

Figura 4. *Schema d'utilizzo delle locazioni contrassegnate dal trattino.*

16514	Numero degli elementi presenti al posto giusto, valutati dal giocatore .
16515	Numero degli elementi presenti, ma al posto sbagliato valutati dal giocatore.
16516-7-8-9, 16520-1-2-3	Locazioni nelle quali viene valutato un codice.
16524-5-6-7	In queste locazioni viene memorizzato il codice del computer quando spetta al giocatore decodificarlo, mentre viene memorizzato il tentativo del computer qua ndo esso deve decodificare il codice del giocatore.
16528	Numero degli elementi presenti al posto giusto, valutati dal computer.
16529	Numero degli elementi presenti, ma al posto sbagliato valutati dal computer.
16760-1	Locazioni nelle quali viene memorizzata la posizione del puntatore che indica la fine di tutte le possibili soluzioni.

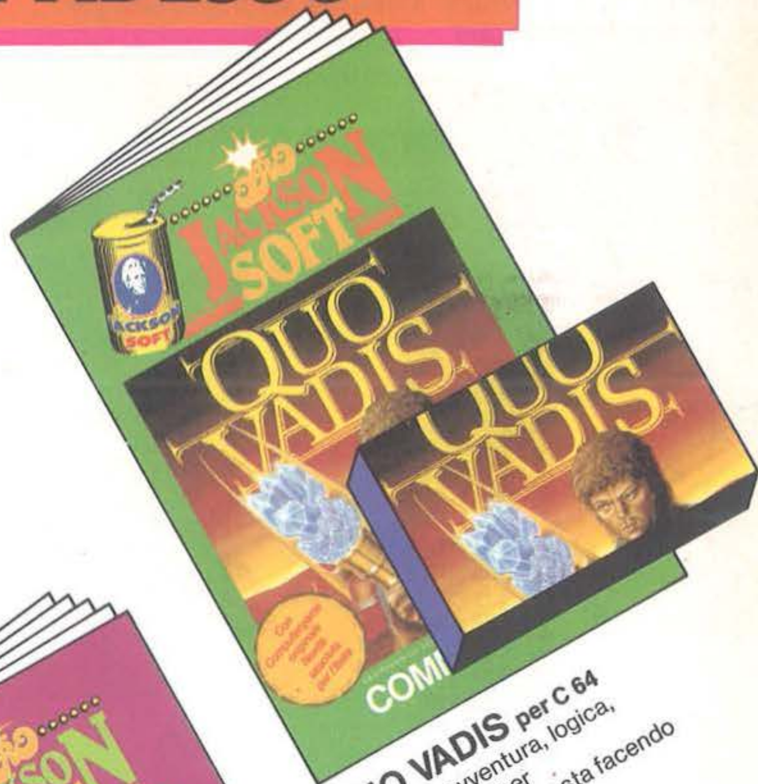
FINALMENTE!

La Softrivista che ti gasa!



QUALCOSA DI SUPER, DI INEDITO, DI IRRESISTIBILE

IL VERO GIOCO COMINCIA ADESSO

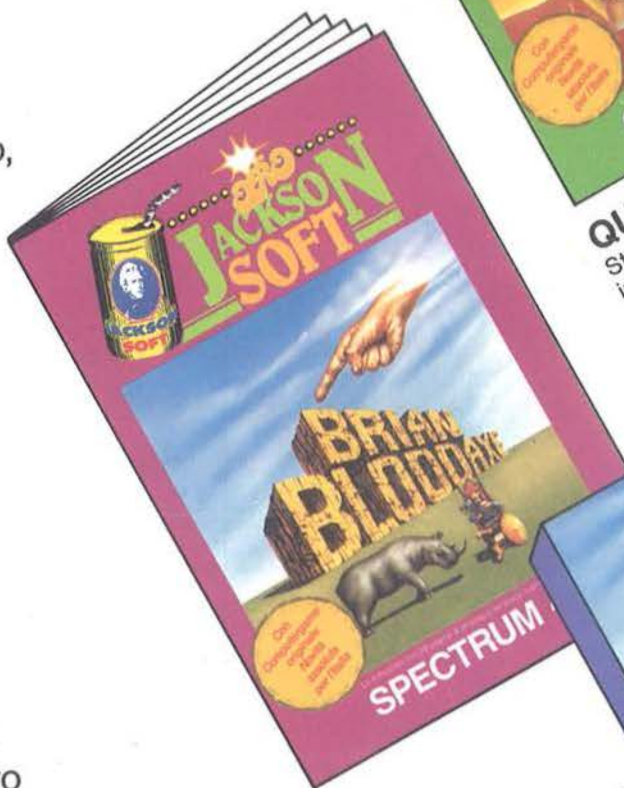


È IN EDICOLA JACKSON SOFT SERIE ORO

La softrivista con i giochi esclusivi per Commodore 64 e Spectrum 48K importati dall'Inghilterra, mai presentati in Italia.
Una sfida Jackson al già visto, al già fatto, al... già registrato.

QUO VADIS per C 64
Strategia, avventura, logica, inganno nel super computergame che sta facendo impazzire l'Inghilterra. Nessuno è ancora riuscito a raggiungere la fine del labirinto. Prova tu!





Ogni mese in edicola

- la softrivista con:
  6 pagine di commento al supergame presentato;
  4 pagine di rubriche;
  22 pagine di listati scelti tra il meglio di quanto pubblicato in fatto di grafica, giochi, utility.
- una cassetta originale con il SUPERGAME del mese

Tutto a sole 10.000 lire

BRIAN BLOODAXE per Spectrum 48K
Un plat forme game che ti condurrà in pieno 21° secolo, alla conquista dei favolosi gioielli della corona inglese. Forza, astuzia, magia e... la colonna sonora di Monty Python.

Corri in edicola, il vero gioco comincia solo adesso e se sei davvero bravo partecipa alla "sfida al campione", utilizzando il tagliando che troverai sull'ultima pagina di copertina di ogni numero.



GRUPPO EDITORIALE JACKSON
SAN FRANCISCO • LONDRA • MILANO

Routine in linguaggio macchina

Figura 5. *Istruzioni Assembler del programma per la gestione dei dati.*

16514	—	0
	—	0
	—	0
	—	0
	—	0
	—	0
	—	0
	—	0
	—	0
	—	0
16524	LD A.16	62 16
	LD (16384).A	50 0 64
16529	LD SP.(16522)	237 123 138 64
	LD IY.16384	253 33 0 64
	CALL 3883	205 43 15
	RET	201
16541	LD (16522).SP	237 115 138 64
	LD HL.16831	33 191 65 vedi articolo
	LD (16520).HL	34 136 64
	LD HL.16519	33 135 64
	LD (HL).0	54 0
	CALL 3875	205 35 15
	LD HL.(16400)	42 16 64
	LD A.H	124
	CP 125	254 125
	JR Z -42	40 213
	LD A.(HL)	126
	INC HL	35
	CP 198	254 198
	JR NZ -10	32 245
	PUSH HL	229
	INC HL	35
	INC HL	35
	LD A.(HL)	126
	CP 2	254 2
	JR Z 3	40 3
	POP HL	225
	JR -16	24 239
	INC HL	35
	LD A.(HL)	126
	LD (16515).A	50 131 64
	INC HL	35
	LD A.(HL)	126
	AND A	167
	JR N Z -12	32 243
	INC HL	35
	LD A.(HL)	126
	LD (16516).A	50 132 64
	INC HL	35
	LD A.(HL)	126
	AND A	167
	JR NZ -22	32 233
	LD HL.(16400)	42 16 64
	LD A.H	124
	CP 125	254 125
	JR Z -87	40 168
	LD A.(HL)	126
	INC HL	35
	CP 92	254 92
	JR NZ -10	32 245
	LD A.(HL)	126
	LD E.A	95
	LD A.(16514)	58 130 64
	CP E	187
	JR NZ -13	32 242
	INC HL	35
	LD A.(HL)	126
	AND A	167
	JR NZ -18	32 237
	INC HL	35
	LD A. (HL)	126
	LD (16518).A	50 134 64
	ADD A.E	131
	DEC A	61
	DEC A	61
	LD E.A	95
	LD A.(16516)	58 132 64
	SUB E	147
	JR C -121	56 134
	NOP	0
	LD (16517).A	50 133 64
	LD A. (HL)	126
	INC HL	35
	PUSH HL	229
	POP IY	253 225
	POP HL	225
	LD DE.6	17 6 0
	ADD HL.DE	25
	AND A	167
	JR Z 6	40 6
	CALL 16693	205 53 65
	JP 16529	195 145 64
	LD A.(16517)	58 133 64
	PUSH HL	229
	PUSH IY	253 229
	CALL 16696	205 56 65
	POP IY	253 225
	POP HL	225
	LD A.(16517)	58 133 64
	DEC A	61
	LD (16517).A	50 133 64
	JR NZ -17	32 238
	JP 16529	195 145 64
16693	LD A.(16518)	58 134 64
	LD C.1	14 1
	LD E.A	95
	ADD HL.DE	25
	LD A.(HL)	126
	LD E.A	95
	LD A.(IY+0)	253 126 0
	CP E	187
	JR Z 14	40 14
	LD A.(16516)	58 132 64
	LD E.A	95
	ADD HL.DE	25
	INC C	12
	LD A.(16515)	58 131 64
	INC A	60
	CP C	185
	JR NZ -20	32 235
	RET	201
	LD A.(16514)	58 130 64
	DEC A	61
	NOP	0
	LD B.A	71
	PUSH HL	229
	PUSH IY	253 229
	LD A.(HL)	126
	LD E.A	95
	LD A.(IY+0)	253 126 0
	INC IY	253 35
	INC HL	35

Seguito figura 5.

	CP E	187
	JR NZ 30	32 30
	DJNZ -12	16 243
	NOP	0
	NOP	0
	LD A.(16519)	58 135 64
	INC A	60
	CP 100	254 100 vedi articolo
	JR NZ 7	32 7
	LD A.15	62 15
	LD (16384).A	50 0 64
	JR 11	24 11
	LD (16519).A	50 135 64
	LD HL.(16520)	42 136 64
	INC HL	35
	LD (16520).HL	34 136 64
	LD (HL).C	113
	POP IY	253 225
	POP HL	225
	JR -68	24 187
16777	—	0
	—	0
16779	LD A.(16519)	58 135 64
	AND A	167
	RET Z	200
	LD B.0	6 0
	LD C.A	79
	LD (16777).BC	237 67 137 65
	LD HL.16832	33 192 65
	LD DE.29000	17 72 113 vedi articolo
	LDIR	237 176
	LD HL.17688	33 24 69
	LD (16744).HL	34 104 65
	CALL 16541	205 157 64
	LD HL.0	33 0 0
	LD (16744).HL	34 104 65
	RET	201
	PUSH BC	197
	LD A.C	121
	LD BC.(16777)	237 75 137 65
	LD HL.29000	33 72 113 vedi articolo
	CPIR	237 177
	POP BC	193
	JR Z -82	40 173
16829	JR -58	24 197

primo carattere del cognome si trova all'inizio di una riga della matrice A$. Possiamo anche, dato il numero telefonico o l'indirizzo, risalire alla persona inserendo uno dei due elementi nella stringa W$ preceduto dalla sua posizione all'interno della riga della matrice. Nel caso in cui tale posizione non fosse nota, il primo elemento di W$ deve essere uguale a Chr$ 0.

Il numero di riga risultante verrà posto a partire dalla locazione 16832; nel caso vi sia più di una riga contenente dei dati uguali (ad esempio due persone con lo stesso cognome), tutti i numeri di queste righe vengono memorizzati sempre a partire dalla locazione 16832; il numero massimo di soluzioni deve comunque essere minore di 100. Questo valore può essere modificato cambiando il contenuto della locazione 16751; si può anche modificare la locazione a partire dalla quale vengono memorizzati i numeri delle righe risultanti, basta modificare adeguatamente le locazioni 16547-8 e 16792-3.

Bisogna inoltre inserire la lunghezza della stringa W$ nella locazione 16514 (Poke 16514, Len W$). La chiamata alla routine avviene tramite Rand Usr 16541, mentre la chiamata per le ricerche successive avviene tramite Rand Usr 16779.

Figura 6 - *Utilizzo delle locazioni di memoria contrassegnate dal trattino nel programma per la gestione dati.*

16514	Lunghezza stringa W$.
16515	Numero di elementi della matrice di stringhe A$.
16516	Lunghezza di ogni stringa della matrice A$.
16517	Differenza tra la lunghezza delle stringhe di A$ e quella di W$
16518	Primo carattere di W$.
16519	Numero di soluzioni trovate.
16520	Locazione di partenza per memorizzare le soluzioni
16521	
16522	Locazione nella quale viene memorizzato lo Stack Pointer.
16523	
16777	Numero di soluzioni trovate per le ricerche successive.
16778	

Per inserire questo programma occorre sempre la routine di caricamento con Let A = 16514 e una Rem di almeno 430 caratteri. I codici da inserire sono quelli della figura 5.

Dalla locazione 16524 a 16540 abbiamo delle istruzioni di controllo in caso di errori e di ritorno al BASIC; da 16541 a 16692 il programma ricerca le stringhe A$ e W$ e ne evidenzia le posizioni, inoltre memorizza separatamente il primo carattere di W$ nella locazione 16518.

La ricerca vera e propria avviene dalla locazione 16693 fino a 16777 mentre la routine di controllo per le ricerche successive si trova a partire da 16779 (per l'utilizzo di alcune locazioni di memoria vedere la figura 6).

Le applicazioni sono innumerevoli: per rendervi conto di ciò provate il programma del listato 5. In esso sono inseriti due interessanti utilizzi: il primo è il vocabolario. Volete ad esempio sapere come si dice in inglese cane basta inserire come posizione della parola 1 o 0 e poi introdurre la parola cane. Sullo schermo apparirà Cane = Dog; si può fare anche il contrario. Il programma può anche essere utilizzato per vedere quali

parole memorizzate contengono determinate lettere o sillabe, per fare cioè delle analisi morfologiche (vedi figura 7b). Il secondo suggerimento che il programma fornisce è per un impiego scientifico; si possono ad esempio memorizzare degli animali e le loro caratteristiche poi vedere quali di essi presentano la particolarità di essere mammiferi e tra i mammiferi vedere quali sono i felini attraverso un'ulteriore analisi sugli animali precedentemente trovati. Un interessante utilizzo di questo tipo consiste nell'inserire le malattie e i loro sintomi, poi, in base a quest'ultimi, risalire alla malattia o malattie che li presentano. In campo informatico si possono realizzare degli assemblatori o dei disassemblatori che associno a una determinata parola un codice numerico e viceversa.

Se il programma BASIC che avete intenzione di realizzare è piuttosto lungo, ricordate che la routine in linguaggio macchina utilizza per le ricerche successive un certo numero di byte a partire dalla locazione 29000; quindi nel caso il vostro programma occupi più di 13 Kbyte, conviene spostare il valore di quella locazione, modificando i valori in 16795-6 e in 16822-3 (ricordiamo che nel primo byte, al posto di 72, va posto un numero che si ottiene da: L-256★INT (L/256) dove L è uguale al valore della nuova locazione. Nel secondo byte, al posto di 113, va posto un nmero pari a Int (L/256). Infine, nel caso vi siano molti dati da analizzare, essi possono venir memorizzati su varie matrici stringa bidimensionali (esempio B$ (100,45); per fare in modo che la routine analizzi la stringa B$ e non A$ basta cambiare il codice della stringa stessa memorizzato nella locazione 16570 (ad A$ corrisponde 198, a B$ 199, a C$ 200...). L'ultimo programma è un'utile routine per i giochi di movimento; essa controlla i tasti schiacciati dal giocatore e sposta in otto direzioni il suo contrassegno, inoltre definisce il campo da gioco con le dimensioni e il carattere voluto e controlla su quale carattere si sta spostando il contrassegno del giocatore.

Le comunicazioni tra programma BASIC e programma in linguaggio macchina avvengono tramite gli elementi di una stringa A$. In questa stringa di otto elementi si inseriscono inizialmente i parametri che definiscono il campo da gioco e la posizione del carattere mosso dal giocatore; gli elementi di questa stringa segnalano la posizione del contrassegno mosso dal giocatore, il carattere su cui tale contrassegno si sta

Listato 5. Esempio d'utilizzo della routine per la gestione dati presentata nella figura 5.

```
   1 REM            Y(M RND GOSUB
 ?▒RND CLEAR 5 RNDLN F?TAN GOSUB
 ?▒RND5▒INKEY$6▒RND5 ▖RNDQ LN 7?E
(RND? RETURN ?CSTR$ FAST 77/ LOA
D 7▣4 NEXT 7▣4 DIM E(RND? RETURN
 ?C▣4 PAUSE 7M▚RND▄XX?U▜RND▒5▚ M
▐RND)▞ ;▣C▞LN PINKEY$?▐RNDU▐RND
FAST CLEAR FAST LN SINKEY$ CLEAR
 LPRINT LPRINT U▐RNDXM▐RND4 INPU
T ?▐RNDU▚RND:▝?;C:U▜RND?;£U▄RNDW
▐4 FOR TAN U▙RNDX ? FAST CLEAR F
AST 77▣42( NEXT  U▖RNDW RETURN
?4▛Y?M RND/"M▖RNDE▒RND76▒RND? CL
EAR LPRINT LPRINT /▣  U▖RND▣COS
▞ ? GOSUB ?▒INKEY$5""INKEY$)?? G
OSUB ▣5/?6?INKEY$LN ▐RND5  6?INK
EY$TAN VAL ? GOSUB ?▒INKEY$5?? G
OSUB ▣AT C▣/VAL ................
................................
................................
................................
...........
  10 DIM A$(14,30)
  20 LET A$(1)="CASA HOME"
  30 LET A$(2)="CANE DOG"
  40 LET A$(3)="SORELLA SISTER"
  50 LET A$(4)="STANZA ROOM"
  60 LET A$(5)="FIUME RIVER"
  70 LET A$(6)="IERI YESTERDAY"
  80 LET A$(7)="BIANCO WHITE"
  90 LET A$(8)="ORANGO MAMMIFERO
PRIMATE"
 100 LET A$(9)="LINCE MAMMIFERO
FELINO"
 110 LET A$(10)="TIGRE MAMMIFERO
FELINO"
 120 LET A$(11)="TASSO MAMMIFERO
MUSTELIDE"
 130 LET A$(12)="TOPO MAMMIFERO
RODITORE"
 140 LET A$(13)="PAVONE UCCELLI
GALLINACEI"
 150 LET A$(14)="RANA ANFIBI ANU
RI"
 160 PRINT "INSERSCI LA POSIZION
E"
 180 PRINT "( 0 SE NON E' NOTA)"
 190 INPUT A
 200 PRINT "INSERISCI I DATI DA
CERCARE"
 210 INPUT U$
 230 LET U$=CHR$ A+U$
 240 POKE 16514,LEN U$
 250 RAND USR 16541
 255 PRINT
 260 FOR I=16832 TO PEEK 16519+1
6831
 270 PRINT A$(PEEK I)
 280 NEXT I
 290 PRINT
 300 PRINT "VUOI FARE UNA ULTERI
ORE ANALISI SUI DATI TROVATI?"
 310 INPUT U$
 320 IF U$="N" OR U$="NO" THEN S
TOP
 330 PRINT "INSERIRE POSIZIONE"
 340 INPUT A
 350 PRINT "INSERIRE ULTERIORI D
ATI"
 360 INPUT U$
 370 LET U$=CHR$ A+U$
 380 POKE 16514,LEN U$
 390 RAND USR 16779
 400 GOTO 260
```

Novità Jackson.

G. Loveday - C. Miere
MISURE ELETTRONICHE E DIAGNOSI DEI GUASTI
Un libro che completa l'istruzione del tecnico elettronico nel campo della rilevazione e delle eliminazione dei guasti dei circuiti elettronici.
Cod. 618P Pag. 368 Lire 34.500

Roberto Doretti
DATA BASE concetti e disegno
Il passato e il presente di un software applicativo fondamentale nell'EDP, per l'utente, il programmatore e il responsabile di archivi elettronici.
Cod. 526P Pag.186 Lire 22.500

B.W. Kernighan, D.M. Ritchie
LINGUAGGIO C
Il testo "ufficiale" sul linguaggio C, scritto da chi l'ha progettato e implementato.
Cod. 541P pag. 216 Lire 21.000

Matteuzzi, Pelizzardi
AMBIENTE UNIX
Un libro che unisce le caratteristiche di un trattato teorico e di un manuale di riferimento, con una completa trattazione del linguaggio C.
Cod. 543P Pag. 198 Lire 19.000

R. Farabone - R. Viano
LOGICA E DIAGRAMMI A BLOCCHI
Il primo libro che affronta in modo sistematico ed esauriente i diagrammi a blocchi, passaggio obbligato per chiunque debba programmare un elaboratore.
Cod. 539A Pag. 392 Lire 37.000

ritagliare (o fotocopiare) e spedire in busta chiusa a:
GRUPPO EDITORIALE JACKSON - Divisione Libri - Via Rosellini, 12 - 20124 Milano

CEDOLA DI COMMISSIONE LIBRARIA

VOGLIATE SPEDIRMI

n° copie	codice	Titolo	Prezzo unitario	Prezzo totale
			Totale	

☐ Pagherò contrassegno al postino il prezzo indicato più **L. 3.000 per contributo fisso spese di spedizione.**

Condizioni di pagamento con esecuzione del contributo spese di spedizione:
☐ Allego assegno della Banca ☐ Allego fotocopia del versamento sul c/c n. 11666203 a voi intestato
N° ______ ☐ Allego fotocopia di versamento su vaglia postale e voi intestato

Nome e Cognome
Via
Cap Città Prov.
Data Firma

Spazio riservato alle Aziende Si richiede l'emissione di fattura
Partita I.V.A. |_|_|_|_|_|_|_|_|_|_|_|

ORDINE MINIMO L. 50.000

Tullio Trincardi
LA CONTABILITÀ COL PERSONAL COMPUTER
Diventare protagonisti diretti nell'utilizzo del computer per le proprie esigenze. È l'obiettivo di un test che affronta il problema dai termini generali fino alle specifiche routine di utilità per la costruzione di programmi gestionali complessi e all'esame di alcuni dei pacchetti più utili alle aziende.
Cod. 570P Pag. 280 Lire 27.000



GRUPPO EDITORIALE JACKSON

La biblioteca che fa testo



Routine in linguaggio macchina

Figura 7a - *Esempio di funzionamento del programma per la gestione dati: qui si è voluto vedere, tra una serie d'animali, quali fossero mammiferi e, tra questi ultimi, quali fossero felini.*

```
INSERSCI LA POSIZIONE
( 0 SE NON E' NOTA) 2
INSERISCI I DATI DA CERCARE
DATO=A

CASA HOME
CANE DOG
TASSO MAMMIFERO MUSTELIDE
PAVONE UCCELLI GALLINACEI
RANA ANFIBIO ANURI

VUOI FARE UNA ULTERIORE ANALISI
SUI DATI TROVATI? SI
INSERIRE POSIZIONE 1
INSERIRE ULTERIORI DATI
DATO ULTERIORE=C

CASA HOME
CANE DOG

VUOI FARE UNA ULTERIORE ANALISI
SUI DATI TROVATI?
```

Figura 7b - *Fra tutte le parole memorizzate, si sono cercate quelle con la A come seconda lettera e poi quelle inizianti con la C.*

```
INSERSCI LA POSIZIONE
( 0 SE NON E' NOTA) 0
INSERISCI I DATI DA CERCARE
DATO=MAMMIFERO

ORANGO MAMMIFERO PRIMATE
LINCE MAMMIFERO FELINO
TIGRE MAMMIFERO FELINO
TASSO MAMMIFERO MUSTELIDE
TOPO MAMMIFERO RODITORE

VUOI FARE UNA ULTERIORE ANALISI
SUI DATI TROVATI? SI
INSERIRE POSIZIONE 0
INSERIRE ULTERIORI DATI
DATO ULTERIORE=FELINO

LINCE MAMMIFERO FELINO
TIGRE MAMMIFERO FELINO

VUOI FARE UNA ULTERIORE ANALIS
SUI DATI TROVATI?
```

Figura 8 - *Istruzioni in linguaggio macchina di una routine da usare nei giochi di movimento.*

16588	—	0
	—	0
	—	0
	—	0
	—	0
	—	0
16594	LD A. 19	62 19
	LD (16384).A	50 0 64
	RET	201
16600	LD HL.(16400)	42 16 64
	LD A.H	124
	CP 125	254 125
	JR Z -13	40 242
	LD A. (HL)	126
	INC HL	35
	CP 198	254 198
	JR NZ -10	32 245
	LD A. (HL)	126
	CP 11	254 11
	JR NZ -10	32 245
	INC HL	35
	INC HL	35
	LD A. (HL)	126
	CP 1	254 1
	JR Z 4	40 4
	DEC HL	43
	DEC HL	43
	JR -21	24 234
	INC HL	35
	INC HL	35
	INC HL	35
	LD (16592).HL	34 208 64
	LD BC.4	1 4 0
	LD DE.16514	17 130 64
	LDIR	237 176
	LD A.(HL)	126
	LD (16519).A	50 135 64
	INC HL	35
	LD A.(HL)	126
	LD (16835).A	50 195 65
	INC HL	35
	LD E. (HL)	94
	INC HL	35
	LD B (HL)	70
	LD D.B	80
	LD (16588).DE	237 83 204 64
	LD D.O	22 0
	LD HL.(16396)	42 12 64
	INC HL	35
	ADD HL.DE	25
	LD DE.33	17 33 0
	ADD HL.DE	25
	DJNZ -2	16 253
	LD (16590).HL	34 206 64
	CALL 16518	205 134 64
	RET	201
16682	LD HL.(16592)	42 208 64
	LD DE.5	17 5 0
	ADD HL.DE	25
	LD (HL).0	54 0
	INC HL	35
	LD (HL).0	54 0
	CALL 699	205 187 2
	LD A.H	124
	ADD A.2	198 2
	JR C 8	56 8
	LD B.H	68
	LD C.L	77
	CALL11981	205 189 7
	LD C.(HL)	78
	JR C 2	56 2
	LD C.0	14 0
	LD A.C	121
16713	CP 37	254 37 tasto "9"
	JR NZ 12	32 12



Seguito figura 8.

	LD DE.5	17 5 0
	LD HL.(16592)	42 208 64
	ADD HL.DE	25
	LD (HL).1	54 1
	JP 16831	195 191 65
16729	CP 28	254 28 tasto "0"
	JR NZ 5	32 5
	LD DE.6	17 6 0
	JR -17	24 238
	LD HL.(16590)	42 204 64
16741	CP 36	254 36 tasto "8"
	JR NZ 2	32 2
	NOP	0
	INC L	44
	CP 33	254 33 tasto "5"
	JR NZ 2	32 2
	NOP	0
	DEC L	45
	CP 34	254 34 tasto "6"
	JR NZ 2	32 2
	NOP	0
	INC H	36
	CP 35	254 35 tasto "7"
	JR NZ 2	32 2
	NOP	0
	DEC H	37
	CP 46	254 46 tasto "I"
	JR NZ 2	32 2
	DEC H	37
	INC L	44
	CP557	254 57 tasto "T"
	JR NZ 2	32 2
	DEC H	37
	DECLL	45
	CP 62	254 62 tasto "Y"
	JR NZ 2	32 2
	INC H	36
	DEC L	45
	CP 58	254 58 tasto "U"
	JR NZ 2	32 2
	INC H	36
	INC L	44
	PUSH HL	229
	LD E.L	93
	LD B.H	68
	LD D.0	22 0
	LD HL.(16396)	42 12 64
	INC HL	35
	ADD HL.DE	25
	LD DE.33	17 3300
	ADD HL.DE	25
	DJNZ -2	16 253
	LD A.(HL)	126
	LD E.A	95
	LD A.(16519)	58 135 64
	CP E	187
	LD A.E	123
	JR NZ 3	32 3
	POP HL	225
	JR 14	24 14
	PUSH HL	229
	LD HL.(16590)	42 206 64
	LD (HL).0	54 0
	POP HL	225
	LD (16590).HL	34 206 64
	POP HL	225
	LD (16588).HL	34 204 64
	LD HL.(16590)	42 206 64
16834	LD (HL).0	54 0
	LD HL.(16592)	42 208 64
	LD DE.16588	17 204 64
	EX HL.DE	235
	LD BC.4	1 4 0
	LDIR	237 176
	LD C.A	79
16849	RET	201

spostando e infine se sono stati premuti i tasti 9 e 0 che possono essere sfruttati per sparare o per operazioni particolari. Così facendo, si evita di usare la funzione Inkey$ e i controlli ad essa connessi, si elimina la stampa del carattere del giocatore e i controlli sulla sua posizione; tutto ciò rende il gioco più veloce.
Per inserire questo programma occorre una Rem di almeno 340 caratteri e la solita routine di caricamento con Let A = 16518; a questo punto dovrete, prima inserire i codici del primo programma presentato sul numero (ci riferiamo al programma che disegna una cornice con un carattere a scelta); dopo di che, bisogna introdurre i codici della figura 8.

La cornice disegnata dal primo programma serve per limitare il campo di gioco e il contrassegno del giocatore non potrà oltrepassarla; se voleste inserire delle strutture (esempio bordi interni di un labirinto) all'interno del campo da gioco tali che il contrassegno non possa oltrepassarle, dovrete costruirle con lo stesso carattere del bordo esterno. La seconda parte del programma serve per rintracciare la stringa A$, valutare gli elementi che la costituiscono, analizzare la situazione della tastiera tramite delle chiamate a due subroutine della ROM (la prima inizia dalla locazione 699, mentre la seconda da 1981), eseguire gli spostamenti relativi ai tasti premuti.

- Tasto "5" il contrassegno si sposta a sinistra.
- Tasto "6" il contrassegno si sposta verso il basso.
- Tasto "7" il contrassegno si sposta verso l'alto.
- Tasto "8" il contrassegno si sposta verso destra.
- Tasto "I" il contrassegno si sposta verso nord-est.
- Tasto "T" il contrassegno si sposta verso nord-ovest.
- Tasto "Y" il contrassegno si sposta verso sud-ovest.
- Tasto "U" il contrassegno si sposta verso sud-est.

La posizione X del carattere mosso dal giocatore si trova nel registro L, mentre la posizione Y si trova nel

Routine in linguaggio macchina

Listato 6. *Esempio d'uso del programma per i giochi in movimento.*

```
   1 REM      :▒- A ▗▝4U▄RND??;( C
LEAR U▙RND?; FAST E£RND7SGN ;U▜R
ND???( UNPLOT F)5 U▐RND?;?( UNPL
OT U▜RND?▐F?( UNPLOT ) TO COPY U
▐RND?▐;?( UNPLOT TAN "£ THEN ???
Y M RNDTAN E(RND? RETURN ?C PAUS
E 7F/ REM 7776SQR RND▝▗ )▙RND GO
SUB █M?INKEY$????? GOSUB ?ATN RN
D- E£RND7;)5 ;( CLEAR 6EXP RNDLN
 ▝▗RNDTAN ESQR RND)▌ ;Q 70 LN █▝?
LEN ▝S▒??LN ▛▀?S▝: ? RETURN 94£)
▐ ESQR RND;Q▝ ?█INKEY$ RETURN 04▐
)▗▝ / INPUT EATN RND RETURN 84▝ G
 RETURN 54▝ H RETURN 64▝ 8 RETUR
N 74▝ 9 RETURN I4▝9G RETURN T4▝9
H RETURN Y4▝8H RETURN U4▝8G FAST
 ??- E£RND7;)5 ;( CLEAR ?4▀ LPRI
NT /: FAST EEXP RNDQ  LPRINT 6EX
P RND LPRINT 6ATN RNDEEXP RNDQ+E
SQR RND)ATN RND FOR ▝▗  GOSUB █?
TAN ::::::::::::::::::::::::::::
::::::::::::::::::::::::::::::::
::
  10 DIM A$(8)
  20 LET A$(1)=CHR$ 2
  30 LET A$(2)=CHR$ 3
  40 LET A$(3)=CHR$ 20
  50 LET A$(4)=CHR$ 15
  60 LET A$(5)=CHR$ 8
  70 LET A$(6)=CHR$ 21
  80 LET A$(7)=CHR$ 10
  90 LET A$(8)=CHR$ 8
 100 LET K=USR 16600
 110 LET K=USR 16682
 115 PRINT AT 1,1;A$
 120 GOTO 110
```

registro H; quindi per spostarsi di una riga si useranno le istruzioni Dec H e Inc H (37 e 36), mentre per spostarsi di una colonna, si useranno le istruzioni Dec L e Inc L (44 e 45). Queste indicazioni possono essere utili nel caso vogliate modificare gli spostamenti e la loro entità; ad esempio, se volete fare in modo che gli spostamenti collegati ai tasti 5, 6, 7, 8 non siano di una casella ma di due, avete a disposizione delle locazioni caratterizzate dall'istruzione Nop (vedi figura 8) nelle quali inserire un ulteriore spostamento utilizzando le quattro istruzioni precedentemente illustrate. Nel caso desideriate fare in modo che il contrassegno si sposti in una direzione indipendentemente dai tasti premuti, bisogna eliminare le istruzioni CP numero tasto e JR NZ 2 relative a quella direzione.

Ad esempio, se vogliamo fare in modo che il contrassegno si sposti sempre di una casella verso destra, dovremo inserire quattro zeri, con delle Poke, nelle locazioni 16741-2-3-4.

La sintassi di funzionamento richiede che la stringa A$ sia inizialmente dimensionata (Dim A$ (8)) con 8 caratteri; i valori da inserire nella stringa sono illustrati nella figura 9.

Per disegnare il bordo e inizializzare la routine occorre eseguire la seguente chiamata: Let K = Usr 16600.

Per giocare bisogna inserire Let K = Usr 16682

Durante il gioco la variabile K assume il valore del carattere su cui il contrassegno si sta spostando. Per quanto riguarda le preziose indicazioni fornite dalla stringa A$, durante il gioco osservate attentamente la figura 9.

Vi è una limitazione per le dimensioni del campo da gioco, e cioè, il bordo superiore non deve occupare la prima riga in alto che potrà essere destinata all'indicazione del nome del programma, o per segnare il punteggio; per il resto valgono le indicazioni dell'articolo precedente. Il programma del listato 6 vi chiarirà meglio le idee. In questo programma lo spigolo in alto a sinistra del bordo ha le coordinate 2,3; la lunghezza dei lati è pari a 20 e 15; il carattere che costituisce il bordo è Chr$ 8; il contrassegno mosso dal giocatore è rappresentato dal carattere Chr$ 21 ed è posto nel punto di coordinate 10,8.

Figura 9 - *Schema d'utilizzo delle locazioni contrassegnate dal trattino del programma di figura 8.*

16588	Coordinata X della posizione del giocatore.
16589	Coordinata Y della posizione del giocatore.
16590-1	Posizione del giocatore nella memoria di schermo.
16592-3	Posizione della stringa A$.

Schema per la definizione iniziale della stringa A$

A$(1) = Chr$	Coordinata X dello spigolo in alto a sinistra del contorno del campo di gioco.
A$(2) = Chr$	Come prima, ma per la Y.
A$(3) = Chr$	Lunghezza lato X.
A$(4) = Chr$	Lunghezza lato Y.
A$(5) = Chr$	Numero del carattere per il contorno.
A$(6) = Chr$	Coordinata X della posizione del giocatore.
A$(7) = Chr$	Coordinata Y della posizione del giocatore.
A$(8) = Chr$	Codice del carattere mosso dal giocatore.

Durante il gioco gli elementi della stringa A$ assumono la seguente funzione

Code A$(1)	Posizione X del giocatore.
Code A$(2)	Posizione Y del giocatore.
Code A$(3)	+ 256 x Code A$(4) = posizione del giocatore nella memoria del calcolatore.
Code A$(6)	Vale 1 se viene premuto il tasto "9", altrimenti assume il valore 0.
Code A$(7)	Come prima, ma per il tasto "0".

Il tuo settimanale





Il primo settimanale di software su carta per il tuo personal computer. Un appuntamento al quale non puoi mancare ogni venerdì in edicola. Nato in edizione unica, per i più diffusi home e personal computer, dal 1 Aprile, allo stesso prezzo, saranno in edicola le versioni dedicate ai prodotti Sinclair (ZX81, ZX Spectrum e QL) e ai prodotti Commodore (VIC20, C16 e C64).
In più un'edizione esclusiva per gli utenti dei personal Apple, gli home computer Texas e i nuovissimi sistemi MSX. Una grossa novità: la possibilità di abbonarsi dal 1 Aprile 1985 al 31 Dicembre 1985 per ricevere 38 numeri del settimanale comodamente a casa con un notevole risparmio. Chi si abbona riceverà ogni 15 giorni 2 numeri di PAPER*Soft* in anticipo rispetto all'uscita in edicola.
Non perdere questa occasione!

Compila subito il coupon qui riportato e spediscilo a:
J.*soft*
V.le Restelli, 5
20124 MILANO

EDITRICE J.soft

Tagliando abbonamento a Paper*Soft* da inviare in busta chiusa a:
J.*soft* - V.le Restelli, 5 - 20124 MILANO

☐ Abbonamento a 38 numeri di Paper*Soft* al prezzo speciale di L. 30.000 anziché L. 38.000

☐ edizione Sinclair (PS01) ☐ edizione Commodore (PC01) ☐ edizione Apple/TI99/MSX (PV01)

☐ contanti allegati ☐ assegno allegato n. ______

☐ ho versato l'importo sul c.c.p. n. 19445204 intestato a J.*soft* - MILANO

Cognome ______ Nome ______

Via ______ Città ______

CAP ______ Prov. ______ Data ______

Firma ______

Dump 1. *Dump di controllo della routine presentata nella figura 1.*

16514 = 0	16563 = 38	16612 = 28		16709 = 3
16515 = 0	16564 = 56	16613 = 218	16661 = 32	16710 = 33
16516 = 0	16565 = 209	16614 = 135	16662 = 2	16711 = 45
16517 = 0	16566 = 14	16615 = 54	16663 = 46	16712 = 36
16518 = 225	16567 = 128	16616 = 57	16664 = 35	16713 = 34
16519 = 62	16568 = 254	16617 = 121	16665 = 254	16714 = 80
16520 = 20	16569 = 127	16618 = 135	16666 = 39	16715 = 65
16521 = 50	16570 = 56	16619 = 135	16667 = 32	16716 = 75
16522 = 0	16571 = 4	16620 = 135	16668 = 2	16717 = 42
16523 = 64	16572 = 14	16621 = 129	16669 = 46	16718 = 130
16524 = 201	16573 = 64	16622 = 129	16670 = 37	16719 = 64
16525 = 42	16574 = 214	16623 = 130	16671 = 254	16720 = 0
16526 = 54	16575 = 128	16624 = 95	16672 = 41	16721 = 0
16527 = 64	16576 = 30	16625 = 35	16673 = 32	16722 = 124
16528 = 34	16577 = 1	16626 = 24	16674 = 3	16723 = 254
16529 = 130	16578 = 50	16627 = 2	16675 = 33	16724 = 44
16530 = 64	16579 = 133	16628 = 89	16676 = 0	16725 = 210
16531 = 62	16580 = 64	16629 = 4	16677 = 44	16726 = 134
16532 = 0	16581 = 121	16630 = 229	16678 = 254	16727 = 64
16533 = 50	16582 = 50	16631 = 58	16679 = 56	16728 = 125
16534 = 132	16583 = 163	16632 = 133	16680 = 32	16729 = 254
16535 = 64	16584 = 65	16633 = 64	16681 = 3	16730 = 64
16536 = 42	16585 = 35	16634 = 254	16682 = 33	16731 = 210
16537 = 16	16586 = 5	16635 = 40	16683 = 0	16732 = 134
16538 = 64	16587 = 40	16636 = 32	16684 = 45	16733 = 64
16539 = 124	16588 = 40	16637 = 7	16685 = 254	16734 = 58
16540 = 254	16589 = 126	16638 = 33	16686 = 54	16735 = 132
16541 = 125	16590 = 254	16639 = 132	16687 = 32	16736 = 64
16542 = 40	16591 = 38	16640 = 64	16688 = 3	16737 = 34
16543 = 231	16592 = 48	16641 = 54	16689 = 33	16738 = 130
16544 = 126	16593 = 35	16642 = 1	16690 = 36	16739 = 64
16545 = 254	16594 = 254	16643 = 24	16691 = 44	16740 = 254
16546 = 70	16595 = 29	16644 = 107	16692 = 254	16741 = 1
16547 = 35	16596 = 218	16645 = 254	16693 = 60	16742 = 40
16548 = 32	16597 = 135	16646 = 53	16694 = 32	16743 = 5
16549 = 245	16598 = 64	16647 = 32	16695 = 3	16744 = 197
16550 = 126	16599 = 214	16648 = 7	16696 = 33	16745 = 205
16551 = 254	16600 = 28	16649 = 33	16697 = 44	16746 = 164
16552 = 0	16601 = 79	16650 = 132	16698 = 37	16747 = 65
16553 = 32	16602 = 35	16651 = 64	16699 = 254	16748 = 193
16554 = 245	16603 = 5	16652 = 54	16700 = 42	16749 = 13
16555 = 35	16604 = 40	16653 = 0	16701 = 32	16750 = 32
16556 = 35	16605 = 22	16654 = 24	16702 = 3	16751 = 221
16557 = 58	16606 = 126	16655 = 95	16703 = 33	16752 = 225
16558 = 168	16607 = 254	16656 = 33	16704 = 37	16753 = 5
16559 = 64	16608 = 38	16657 = 0	16705 = 45	16754 = 194
16560 = 71	16609 = 48	16658 = 0	16706 = 254	16755 = 177
16561 = 126	16610 = 17	16659 = 254	16707 = 55	16756 = 64
16562 = 254	16611 = 214	16660 = 38	16708 = 32	16757 = 201

Dump 2. *Dump di controllo per inserire e verificare i codici del listato Assembly della figura 2.*

16514 = 0	16527 = 0	16540 = 6	16553 = 32	16566 = 33
16515 = 0	16528 = 0	16541 = 112	16554 = 240	16567 = 132
16516 = 0	16529 = 0	16542 = 35	16555 = 29	16568 = 64
16517 = 0	16530 = 33	16543 = 115	16556 = 32	16569 = 17
16518 = 0	16531 = 168	16544 = 35	16557 = 235	16570 = 136
16519 = 0	16532 = 97	16545 = 114	16558 = 16	16571 = 64
16520 = 0	16533 = 6	16546 = 35	16559 = 231	16572 = 14
16521 = 0	16534 = 6	16547 = 113	16560 = 54	16573 = 0
16522 = 0	16535 = 30	16548 = 35	16561 = 117	16574 = 229
16523 = 0	16536 = 6	16549 = 13	16562 = 34	16575 = 6
16524 = 0	16537 = 22	16550 = 32	16563 = 120	16576 = 4
16525 = 0	16538 = 6	16551 = 245	16564 = 65	16577 = 26
16526 = 0	16539 = 14	16552 = 21	16565 = 201	16578 = 190

Seguito Dump 2.

```
16579 = 32
16580 = 7
16581 = 12
16582 = 54
16583 = 0
16584 = 235
16585 = 54
16586 = 8
16587 = 235
16588 = 35
16589 = 19
16590 = 16
16591 = 241
16592 = 121
16593 = 50
16594 = 144
16595 = 64
16596 = 225
16597 = 14
16598 = 0
16599 = 6
16600 = 4
16601 = 126
16602 = 229
16603 = 30
16604 = 4
16605 = 33
16606 = 136
16607 = 64
16608 = 190
16609 = 32
16610 = 5
16611 = 12
16612 = 54
16613 = 8
16614 = 24
16615 = 4
```

```
16616 = 35
16617 = 29
16618 = 32
16619 = 244
16620 = 225
16621 = 35
16622 = 16
16623 = 233
16624 = 121
16625 = 50
16626 = 145
16627 = 64
16628 = 201
16629 = 33
16630 = 140
16631 = 64
16632 = 17
16633 = 132
16634 = 64
16635 = 1
16636 = 4
16637 = 0
16638 = 237
16639 = 176
16640 = 205
16641 = 182
16642 = 64
16643 = 201
16644 = 33
16645 = 168
16646 = 97
16647 = 126
16648 = 254
16649 = 117
16650 = 32
16651 = 6
16652 = 62
```

```
16653 = 16
16654 = 50
16655 = 0
16656 = 64
16657 = 201
16658 = 229
16659 = 1
16660 = 4
16661 = 0
16662 = 33
16663 = 140
16664 = 64
16665 = 17
16666 = 132
16667 = 64
16668 = 237
16669 = 176
16670 = 225
16671 = 17
16672 = 136
16673 = 64
16674 = 1
16675 = 4
16676 = 0
16677 = 237
16678 = 176
16679 = 229
16680 = 205
16681 = 182
16682 = 64
16683 = 225
16684 = 58
16685 = 130
16686 = 64
16687 = 79
16688 = 58
```

```
16689 = 144
16690 = 64
16691 = 185
16692 = 40
16693 = 21
16694 = 1
16695 = 4
16696 = 0
16697 = 43
16698 = 235
16699 = 42
16700 = 120
16701 = 65
16702 = 43
16703 = 237
16704 = 184
16705 = 35
16706 = 19
16707 = 54
16708 = 117
16709 = 34
16710 = 120
16711 = 65
16712 = 235
16713 = 24
16714 = 10
16715 = 58
16716 = 131
16717 = 64
16718 = 79
16719 = 58
16720 = 145
16721 = 64
16722 = 185
16723 = 32
16724 = 225
16725 = 126
```

```
16726 = 254
16727 = 117
16728 = 32
16729 = 184
16730 = 17
16731 = 168
16732 = 97
16733 = 237
16734 = 82
16735 = 26
16736 = 254
16737 = 117
16738 = 202
16739 = 12
16740 = 65
16741 = 203
16742 = 61
16743 = 203
16744 = 61
16745 = 203
16746 = 60
16747 = 48
16748 = 2
16749 = 203
16750 = 245
16751 = 203
16752 = 60
16753 = 48
16754 = 2
16755 = 203
16756 = 253
16757 = 68
16758 = 77
16759 = 201
16760 = 0
16761 = 0
```

Dump 3. *Dump della routine della figura 5.*

```
16514 = 0
16515 = 0
16516 = 0
16517 = 0
16518 = 0
16519 = 0
16520 = 0
16521 = 0
16522 = 0
16523 = 0
16524 = 62
16525 = 16
16526 = 50
16527 = 0
16528 = 64
16529 = 237
16530 = 123
16531 = 138
16532 = 64
16533 = 253
16534 = 33
16535 = 0
16536 = 64
16537 = 205
```

```
16538 = 43
16539 = 15
16540 = 201
16541 = 237
16542 = 115
16543 = 138
16544 = 64
16545 = 33
16546 = 191
16547 = 65
16548 = 34
16549 = 136
16550 = 64
16551 = 33
16552 = 135
16553 = 64
16554 = 54
16555 = 0
16556 = 205
16557 = 35
16558 = 15
16559 = 42
16560 = 16
16561 = 64
```

```
16562 = 124
16563 = 254
16564 = 125
16565 = 40
16566 = 213
16567 = 126
16568 = 35
16569 = 254
16570 = 198
16571 = 32
16572 = 245
16573 = 229
16574 = 35
16575 = 35
16576 = 126
16577 = 254
16578 = 2
16579 = 40
16580 = 3
16581 = 225
16582 = 24
16583 = 239
16584 = 35
16585 = 126
```

```
16586 = 50
16587 = 131
16588 = 64
16589 = 35
16590 = 126
16591 = 167
16592 = 32
16593 = 243
16594 = 35
16595 = 126
16596 = 50
16597 = 132
16598 = 64
16599 = 35
16600 = 126
16601 = 167
16602 = 32
16603 = 233
16604 = 42
16605 = 16
16606 = 64
16607 = 124
16608 = 254
16609 = 125
```

```
16610 = 40
16611 = 168
16612 = 126
16613 = 35
16614 = 254
16615 = 92
16616 = 32
16617 = 245
16618 = 126
16619 = 95
16620 = 58
16621 = 130
16622 = 64
16623 = 187
16624 = 32
16625 = 242
16626 = 35
16627 = 126
16628 = 167
16629 = 32
16630 = 237
16631 = 35
16632 = 126
16633 = 50
```

Seguito Dump 3.

```
16634 = 134
16635 = 64
16636 = 131
16637 = 61
16638 = 61
16639 = 95
16640 = 58
16641 = 132
16642 = 64
16643 = 147
16644 = 56
16645 = 134
16646 = 0
16647 = 50
16648 = 133
16649 = 64
16650 = 126
16651 = 35
16652 = 229
16653 = 253
16654 = 225
16655 = 225
16656 = 17
16657 = 6
16658 = 0
16659 = 25
16660 = 167
16661 = 40
16662 = 6
16663 = 205
16664 = 53
16665 = 65
16666 = 195
16667 = 145
16668 = 64
16669 = 58
16670 = 133
16671 = 64
16672 = 229
16673 = 253

16674 = 229
16675 = 205
16676 = 55
16677 = 65
16678 = 253
16679 = 225
16680 = 225
16681 = 58
16682 = 133
16683 = 64
16684 = 61
16685 = 50
16686 = 133
16687 = 64
16688 = 32
16689 = 238
16690 = 195
16691 = 145
16692 = 64
16693 = 58
16694 = 134
16695 = 64
16696 = 14
16697 = 1
16698 = 95
16699 = 25
16700 = 126
16701 = 95
16702 = 253
16703 = 126
16704 = 0
16705 = 187
16706 = 40
16707 = 14
16708 = 58
16709 = 132
16710 = 64
16711 = 95
16712 = 25

16713 = 12
16714 = 58
16715 = 131
16716 = 64
16717 = 60
16718 = 135
16719 = 32
16720 = 235
16721 = 201
16722 = 58
16723 = 130
16724 = 64
16725 = 61
16726 = 0
16727 = 71
16728 = 229
16729 = 253
16730 = 229
16731 = 126
16732 = 95
16733 = 253
16734 = 126
16735 = 0
16736 = 253
16737 = 35
16738 = 35
16739 = 187
16740 = 32
16741 = 30
16742 = 16
16743 = 243
16744 = 0
16745 = 0
16746 = 58
16747 = 135
16748 = 64
16749 = 60
16750 = 254
16751 = 100
16752 = 32

16753 = 7
16754 = 62
16755 = 15
16756 = 50
16757 = 0
16758 = 64
16759 = 24
16760 = 11
16761 = 50
16762 = 135
16763 = 64
16764 = 42
16765 = 136
16766 = 64
16767 = 35
16768 = 34
16769 = 136
16770 = 64
16771 = 113
16772 = 253
16773 = 225
16774 = 225
16775 = 24
16776 = 187
16777 = 0
16778 = 0
16778 = 0
16779 = 58
16780 = 135
16781 = 64
16782 = 167
16783 = 200
16784 = 6
16785 = 0
16786 = 79
16787 = 237
16788 = 67
16789 = 137
16790 = 65

16791 = 33
16792 = 192
16793 = 65
16794 = 17
16795 = 72
16796 = 113
16797 = 237
16798 = 176
16799 = 33
16800 = 24
16801 = 69
16802 = 34
16803 = 104
16804 = 65
16805 = 205
16806 = 157
16807 = 64
16808 = 33
16809 = 0
16810 = 0
16811 = 34
16812 = 104
16813 = 65
16814 = 201
16815 = 197
16816 = 121
16817 = 237
16818 = 75
16819 = 137
16820 = 65
16821 = 33
16822 = 72
16823 = 113
16824 = 237
16825 = 177
16826 = 193
16827 = 40
16828 = 173
16829 = 24
16830 = 197
```

Dump 4. *Dump della routine per i giochi in movimento visibile in figura 8.*

```
16588 = 0
16589 = 0
16590 = 0
16591 = 0
16592 = 0
16593 = 0
16594 = 62
16595 = 19
16596 = 50
16597 = 0
16598 = 64
16599 = 201
16600 = 42
16601 = 16
16602 = 64
16603 = 124
16604 = 254
16605 = 125
16606 = 40
16607 = 242

16608 = 126
16609 = 35
16610 = 254
16611 = 198
16612 = 32
16613 = 245
16614 = 126
16615 = 254
16616 = 11
16617 = 32
16618 = 245
16619 = 35
16620 = 35
16621 = 126
16622 = 254
16623 = 1
16624 = 40
16625 = 4
16626 = 43
16627 = 43

16628 = 24
16629 = 234
16630 = 35
16631 = 35
16632 = 35
16633 = 34
16634 = 208
16635 = 64
16636 = 1
16637 = 4
16638 = 0
16639 = 17
16640 = 130
16641 = 64
16642 = 237
16643 = 176
16644 = 126
16645 = 50
16646 = 135
16647 = 64

16648 = 35
16649 = 126
16650 = 50
16651 = 195
16652 = 65
16653 = 35
16654 = 94
16655 = 35
16656 = 70
16657 = 80
16658 = 237
16659 = 83
16660 = 204
16661 = 64
16662 = 22
16663 = 0
16664 = 42
16665 = 12
16666 = 64
16667 = 35

16668 = 25
16669 = 17
16670 = 33
16671 = 0
16672 = 25
16673 = 16
16674 = 253
16675 = 34
16676 = 206
16677 = 64
16678 = 205
16679 = 134
16680 = 64
16681 = 201
16682 = 42
16683 = 208
16684 = 64
16685 = 17
16686 = 5
16687 = 0
```



Routine in linguaggio macchina

Seguito Dump 4.

16688 = 25	16720 = 42	16753 = 254	16785 = 32	16817 = 229
16689 = 54	16721 = 208	16754 = 34	16786 = 2	16818 = 42
16690 = 0	16722 = 64	16755 = 32	16787 = 36	16819 = 206
16691 = 35	16723 = 25	16756 = 2	16788 = 44	16820 = 64
16692 = 54	16724 = 54	16757 = 0	16789 = 229	16821 = 54
16693 = 0	16725 = 1	16758 = 36	16790 = 93	16822 = 0
16694 = 205	16726 = 195	16759 = 254	16791 = 68	16823 = 225
16695 = 187	16727 = 191	16760 = 35	16792 = 22	16824 = 34
16696 = 2	16728 = 65	16761 = 32	16793 = 0	16825 = 206
16697 = 124	16729 = 254	16762 = 2	16794 = 42	16826 = 64
16698 = 198	16730 = 28	16763 = 0	16795 = 12	16827 = 225
16699 = 2	16731 = 32	16764 = 37	16796 = 64	16828 = 34
16700 = 56	16732 = 5	16765 = 254	16797 = 35	16829 = 204
16701 = 8	16733 = 17	16766 = 45	16798 = 25	16830 = 64
16702 = 68	16734 = 6	16767 = 32	16799 = 17	16831 = 42
16703 = 77	16735 = 0	16768 = 2	16800 = 33	16832 = 206
16704 = 205	16736 = 24	16769 = 37	16801 = 0	16833 = 64
16705 = 189	16737 = 238	16770 = 44	16802 = 25	16834 = 54
16706 = 7	16738 = 42	16771 = 254	16803 = 16	16835 = 0
16707 = 78	16739 = 204	16772 = 57	16804 = 253	16836 = 42
16708 = 56	16740 = 64	16773 = 32	16805 = 126	16837 = 208
16709 = 2	16741 = 254	16774 = 2	16806 = 95	16838 = 64
16710 = 14	16742 = 35	16775 = 37	16807 = 58	16839 = 17
16711 = 0	16743 = 32	16776 = 45	16808 = 135	16840 = 204
16712 = 121	16744 = 2	16777 = 254	16809 = 64	16841 = 64
16713 = 254	16745 = 0	16778 = 62	16810 = 187	16842 = 235
16714 = 37	16746 = 44	16779 = 32	16811 = 123	16843 = 1
16715 = 32	16747 = 254	16780 = 2	16812 = 32	16844 = 4
16716 = 12	16748 = 33	16781 = 36	16813 = 3	16845 = 0
16717 = 17	16749 = 32	16782 = 45	16814 = 225	16846 = 237
16718 = 5	16750 = 2	16783 = 254	16815 = 24	16847 = 176
16719 = 0	16751 = 0	16784 = 58	16816 = 14	16848 = 79
	16752 = 45			16849 = 201

Guida per l'input dei programmi versioni VIC 20 e C 64

Notate che i listati contengono "parole" racchiuse tra parentesi graffe { } . Tali parole rappresentano caratteri di controllo come mostrato nel sottostante riquadro. Se sono precedute da un numero, questo indica il numero di volte che quel tasto deve essere premuto. Se il simbolo è sottolineato deve essere premuto contemporaneamente a SHIFT mentre se è racchiuso da [< >] deve essere premuto contemporaneamente al tasto COMMODORE. Inoltre, se tra parentesi si trova un carattere alfabetico "solitario", questo dovrà essere premuto contemporaneamente al tasto CONTROL.

Con questo sistema di codifica, sarà molto più agevole copiare i listati senza faticose e dubbie interpretazioni di caratteri grafici e di controllo di cursore o dei colori.

Quando leggete:	Premete:	Vedrete:	Quando leggete:	Premete:	Vedrete:	Quando leggete:	Premete:	Vedrete:
{CLR}	SHIFT CLR HOME		{CYN}	CTRL		[<7>]	C=	
{HOME}	CLR HOME		{PUR}	CTRL		[<8>]	C=	
{SU}	SHIFT CRSR		{GRN}	CTRL		{F1}	F1	
{GIU'}	CRSR		{BLU}	CTRL		{F2}	F2	
{SIN}	SHIFT CRSR		{YEL}	CTRL		{F3}	F3	
{DES}	CRSR		[<1>]	C=		{F4}	F4	
{RVS}	CTRL		[<2>]	C=		{F5}	F5	
{OFF}	CTRL		[<3>]	C=		{F6}	F6	
{BLK}	CTRL		[<4>]	C=		{F7}	F7	
{WHT}	CTRL		[<5>]	C=		{F8}	F8	
{RED}	CTRL		[<6>]	C=				

PERSONAL SOFTWARE

Caccia al tredici

Anche il vostro Texas TI99 può vincere al Totocalcio

di *Carlo Aceti*

Caccia al tredici è un programma che aiuta nella stesura della schedina del Totocalcio, soprattutto se non si è degli esperti in materia. Prima di passare al listato, spenderemo due parole per spiegare cosa fare dopo il Run. Anzitutto, dato che il programma per funzionare ha bisogno di alcuni dati, bisognerà dapprima procurarsi un giornale specializzato, dal quale si potranno attingere le informazioni desiderate.
Dopo il Run vedremo l'intestazione, quindi apparirà il titolo, infine verranno fatte le prime domande, di carattere generale, sulla schedina in questione.
Una di queste domande, "Quante doppie max vuoi?", necessita forse di un chiarimento.

Con essa viene richiesto il numero massimo di doppie che desiderate; non è detto però che vi saranno date tutte: ne otterrete tante quante saranno state elaborate dal computer; solo nel caso che esse siano più di quelle richieste, il computer provvederà ad eliminare l'eccedenza.
Se dopo l'ultima domanda, "Tutto OK?", la risposta è affermativa (basta solo un Enter) lo schermo cambierà colore e si passerà alla terza pagina, ossia alle domande relative alle squadre che giocano in casa.
Cioè:

Listato 1. *Il programma Totocalcio.*

```
10 CALL CLEAR :: CALL SCREEN(11)
:: PRINT " CACCIA AL 13": : :
20 PRINT " PROGRAMMA PER TI-99/4
A": : :
30 PRINT " EXT.BASIC ": : :
40 PRINT " REALIZZATO DA ACETI C
ARLO": : :
50 PRINT :: PRINT
60 CALL RITARDO(1000)
70 PRINT "$$$    BUONA FORTUNA
 $$$": : :
80 CALL RITARDO(1000)
90 PRINT :: PRINT :: PRINT
100 CALL TITOLO
110 CALL SCREEN(11)
120 OPTION BASE 1
130 DIM A$(26,2),RIS$(13,1),B$(1
3,1):: A=4 :: K=0 :: AA=4 :: C=4
 :: CC=4 :: Q=1
140 CALL CLEAR
150 DISPLAY AT(3,5):"RISPONDI AL
LE DOMANDE"
160 CALL LINO(5,2,30):: CALL LIN
O(10,2,30):: CALL LINO(14,2,30):
: CALL LINO(18,2,30)
170 DISPLAY AT(6,5):"QUANTE SONO
 LE PAR-" :: DISPLAY AT(7,5):"TI
TE CHE SI GIOCANO" :: DISPLAY AT
(8,5):"IN SERIE A,B,C1? >"
180 ACCEPT AT(8,24)VALIDATE(NUME
RIC)SIZE(2):TT :: ON WARNING NEX
T :: TOT=TT
190 DISPLAY AT(11,5):"QUANTE DOP
PIE" :: DISPLAY AT(12,5):"MASSIM
O VUOI?" :: ACCEPT AT(12,24)VALI
DATE(NUMERIC)SIZE(2):X
200 ON WARNING NEXT
210 DISPLAY AT(15,5):"CHE NUMERO
 DI" :: DISPLAY AT(16,5):"PARTIT
E SIAMO ?" :: ACCEPT AT(16,24)VA
LIDATE(NUMERIC)SIZE(2):GIO :: GI
O1=GIO
220 ON WARNING NEXT
230 GIO=GIO/100
240 DISPLAY AT(20,6):"TUTTO OK<S
/N)?" :: ACCEPT AT(20,20)VALIDAT
E("NS")SIZE(-1):G$ :: CALL CLEAR

250 IF G$="N" OR G$="N" THEN GOT
O 150
260 CALL SCREEN(16)
270 F2$="IN CASA"
280 F3$="TRASF. "
290 F$="VIT. PAR. SCO. DIF."
300 FOR I=1 TO TOT
310 CALL LINO(3,1,32):: CALL LIN
O(19,1,32)
320 DISPLAY AT(1,2):F2$&" "&F$
```

Caccia al tredici

Seguito listato Totocalcio.

```
330 CALL NUM
340 ACCEPT AT(A,4)VALIDATE(UALPH
A)SIZE(5):A$(I,1):: ACCEPT AT(A,
10)VALIDATE(NUMERIC)SIZE(2):V
350 ON WARNING NEXT
360 ACCEPT AT(A,15)VALIDATE(NUME
RIC)SIZE(2):P :: ACCEPT AT(A,20)
VALIDATE(NUMERIC)SIZE(2):S
370 ACCEPT AT(A,25)VALIDATE(NUME
RIC)SIZE(2):D
380 RI=(2*V)+(1*P)-(1.1*S)-(0.3*
D)
390 IF GIO1>(V+P+S)THEN RI=(RI/(
V+P+S))*GIO1
400 ON WARNING NEXT
410 A=A+1 :: IF A=17 THEN CALL H
CHAR(4,1,32,480):: A=4
420 A$(I,2)=STR$(RI)
430 NEXT I
440 TOT=TOT+1
450 IF TT=0 THEN TOT=1
460 FOR I=TOT TO 13
470 DISPLAY AT(1,2):F2$&" "&F$
480 CALL NUM
490 CALL LINO(3,1,32):: CALL LIN
O(19,1,32)
500 ACCEPT AT(A,4)VALIDATE(UALPH
A)SIZE(5):A$(I,1):: ACCEPT AT(A,
10)VALIDATE(NUMERIC)SIZE(2):VV
510 ACCEPT AT(A,15)VALIDATE(NUME
RIC)SIZE(2):PP :: ACCEPT AT(A,20
)VALIDATE(NUMERIC)SIZE(2):SS
520 ACCEPT AT(A,25)VALIDATE(NUME
RIC)SIZE(2):DD
530 A=A+1 :: IF A=17 THEN CALL H
CHAR(4,1,32,480):: A=4
540 ON WARNING NEXT
550 RI2=((VV*1)+(PP*.5)-(SS*.7))
/(DD*.3):: A$(I,2)=STR$(RI2)
560 NEXT I
570 IF TOT<>0 THEN TOT=TOT-1
580 FOR I=14 TO 13+TOT
590 IF TT=0 THEN GOTO 710
600 DISPLAY AT(1,2):F3$&" "&F$
610 CALL NUM
620 ACCEPT AT(A,4)VALIDATE(UALPH
A)SIZE(5):A$(I,1):: ACCEPT AT(A,
10)VALIDATE(NUMERIC)SIZE(2):V1
630 ON WARNING NEXT
640 ACCEPT AT(A,15)VALIDATE(NUME
RIC)SIZE(2):P1 :: ACCEPT AT(A,20
)VALIDATE(NUMERIC)SIZE(2):S1
650 ACCEPT AT(A,25)VALIDATE(NUME
RIC)SIZE(2):D1
660 RI1=(2.1*V1)+(1.5*P1)-(.9*S1
)-(0.3*D1)
670 IF GIO1>(V1+P1+S1)THEN RI1=(
RI1/(V1+P1+S1))*GIO1
680 ON WARNING NEXT
690 A=A+1 :: A$(I,2)=STR$(RI1)
700 NEXT I
710 IF TT=0 THEN TOT=1 ELSE TOT=
TOT+1
720 FOR I=13+TOT TO 26
730 DISPLAY AT(1,2):F3$&" "&F$
740 CALL NUM
750 ACCEPT AT(A,4)VALIDATE(UALPH
A)SIZE(5):A$(I,1):: ACCEPT AT(A,
10)VALIDATE(NUMERIC)SIZE(2):VV
760 ACCEPT AT(A,15)VALIDATE(NUME
RIC)SIZE(2):PP :: ACCEPT AT(A,20
)VALIDATE(NUMERIC)SIZE(2):SS
770 ACCEPT AT(A,25)VALIDATE(NUME
RIC)SIZE(2):DD
780 ON WARNING NEXT
790 RI2=((VV*1)+(PP*.5)-(SS*.7))
/(DD*.3):: A=A+1 :: A$(I,2)=STR$
(RI2)
800 NEXT I
810 CALL CLEAR
820 CALL LINV(4,11,13):: CALL LI
NV(4,19,13)
830 CALL LINO(3,1,32):: CALL LIN
O(18,1,32)
840 CALL TITF
850 FOR I=1 TO 13
860 DISPLAY AT(C,1)SIZE(7):A$(I,
1)
870 C=C+1 :: NEXT I
880 FOR D=14 TO 26
890 DISPLAY AT(CC,10)SIZE(7):A$(
D,1)
900 CC=CC+1 :: NEXT D
910 FOR IN=1 TO 13
920 SC1=VAL(A$(IN,2))
930 SC2=VAL(A$(IN+13,2))
940 GOSUB 1000
950 NEXT IN
960 GOSUB 1220
970 GOSUB 1350
980 CALL KEY(0,T,S):: IF S=0 THE
N 980 :: IF T=67 THEN GOTO 110
990 END
1000 REM CONTEGGIO
1010 IF SC1>SC2 THEN GOTO 1030
1020 IF SC1<SC2 THEN GOTO 1100
1030 IF (SC1-SC2)>=INT((.05+GIO)
```

Seguito listato Totocalcio.

```
*SC2)THEN RI$="2"
1040 IF (SC1-SC2)>=INT((.10+GIO)
*SC2)THEN RI$="2X"
1050 IF (SC1-SC2)>=INT((.19+GIO)
*SC2)THEN RI$="X1"
1060 IF (SC1-SC2)>=INT((.29+GIO)
*SC2)THEN RI$="X"
1070 IF (SC1-SC2)>=INT((.35+GIO)
*SC2)THEN RI$="1X"
1080 IF (SC1-SC2)>=INT((.45+GIO)
*SC2)THEN RI$="1"
1090 GOTO 1160
1100 IF (SC2-SC1)>=INT((.05+GIO)
*SC1)THEN RI$="1"
1110 IF (SC2-SC1)>=INT((.10+GIO)
*SC1)THEN RI$="1X"
1120 IF (SC2-SC1)>=INT((.19+GIO)
*SC1)THEN RI$="X"
1130 IF (SC2-SC1)>=INT((.29+GIO)
*SC1)THEN RI$="X1"
1140 IF (SC2-SC1)>=INT((.35+GIO)
*SC1)THEN RI$="2X"
1150 IF (SC2-SC1)>=INT((.45+GIO)
*SC1)THEN RI$="2"
1160 IF LEN(RI$)=2 THEN K=K+1
1170 IF K>X THEN RI$=SEG$(RI$,1,
1)ELSE RI$=RI$
1180 RIS$(Q,1)=RI$ :: Q=Q+1
1190 DISPLAY AT(AA,20)SIZE(2):RI
$
1200 AA=AA+1
1210 RETURN
1220 REM CASUALITA`
1230 AA=4
1240 FOR I=1 TO 13
1250 RANDOMIZE
1260 S=INT(4*RND)+1
1270 ON S GOSUB 1290,1300,1310,1
320
1280 GOTO 1330
1290 B$(I,1)="1" :: RETURN
1300 B$(I,1)="1X" :: RETURN
1310 B$(I,1)="2" :: RETURN
1320 B$(I,1)="X" :: RETURN
1330 DISPLAY AT(AA,23)SIZE(2):B$
(I,1)
1340 AA=AA+1 :: NEXT I :: RETURN
1350 AA=4
1360 REM MEDIA
1370 RANDOMIZE
1380 FOR I=1 TO 13
1390 MED=INT(2*RND)+1
1400 ON MED GOSUB 1420,1430
1410 GOTO 1440
1420 MED$=RIS$(I,1):: RETURN
1430 MED$=B$(I,1):: RETURN
1440 IF LEN(MED$)=2 THEN W=W+1
1450 IF W>X THEN MED$=SEG$(MED$,
1,1)ELSE MED$=MED$
1460 DISPLAY AT(AA,26)SIZE(2):ME
D$ :: AA=AA+1
1470 NEXT I
1480 RETURN
1490 SUB TITOLO :: CALL CLEAR ::
 CALL COLOR(2,5,5)
1500 A5$=" + ++  +  + +++" :: A6
$="+++  ++++  ++++"
1510 A1$="++++  +  +++" :: A2$="
++++ +++++ +" :: A3$="+  +  +  +
  " :: A4$="+  +  +  +++"
1520 FOR I=1 TO 4 :: DISPLAY AT(
I+7,5):SEG$(A1$,I*3-2,3):: NEXT
I
1530 CALL SOUND(10,1000,4)
1540 FOR I=1 TO 4 :: DISPLAY AT(
I+7,9):SEG$(A2$,I*3-2,3):: NEXT
I
1550 CALL SOUND(10,1000,4)
1560 FOR I=1 TO 4 :: DISPLAY AT(
```

- il nome della squadra;
- le vittorie in casa;
- i pareggi in casa;
- le sconfitte in casa;
- coefficiente di difficoltà (in relazione all'abilità della squadra avversaria).

Quest'ultima domanda costituisce un tocco personale riservato ai più esperti. Potete "penalizzare" una squadra assegnandole un coefficiente di difficoltà, variabile da 0 a 99, nell'affrontare la squadra avversaria.

Dopo le tredici squadre che giocano in casa, automaticamente si passerà a quelle che giocano in trasferta.

Due precisazioni:

- nel listato il nome della squadra non potrà essere lungo più di 5 lettere;
- quando scrivete il nome delle squadre, ricordate che il computer confronterà la prima delle squadre che giocano in casa con la prima di quelle che giocano in trasferta, procedendo in modo analogo per tredici volte. Sarebbe perciò consigliabile seguire l'ordine indicato dalla schedina.

L'ultima pagina è completamente gestita dal nostro TI99.

Vedremo apparire i nomi delle squadre che si affrontano e tre tipi di risultati:

1) logico: ossia il logico risultato scaturito dai dati che sono stati immessi;
2) random: cioè scelta casuale ad opera del computer;
3) misto: dove casualmente vengono mischiati i due tipi di risultati precedenti, nella speranza di poter vincere più facilmente.

Grazie ad un key, il video rimane fermo per tutto il tempo che vogliamo: premendo la C maiuscola torneremo alle prime domande; premendo qualsiasi altro tasto fermeremo il programma.

Ed ora il listato.



Caccia al tredici

Seguito listato Totocalcio.

```
I+7,13):SEG$(A1$,I*3-2,3):: NEXT
 I
1570 CALL SOUND(10,1000,4)
1580 FOR I=1 TO 4 :: DISPLAY AT(
I+7,17):SEG$(A1$,I*3-2,3):: NEXT
 I
1590 CALL SOUND(10,1000,4)
1600 FOR I=1 TO 4 :: DISPLAY AT(
I+7,21):SEG$(A3$,I*3-2,3):: NEXT
 I
1610 CALL SOUND(10,1000,4)
1620 FOR I=1 TO 4 :: DISPLAY AT(
I+7,23):SEG$(A2$,I*3-2,3):: NEXT
 I
1630 CALL SOUND(10,1000,4)
1640 FOR I=1 TO 4 :: DISPLAY AT(
I+13,7):SEG$(A2$,I*3-2,3):: NEXT
 I
1650 CALL SOUND(10,1000,4)
1660 FOR I=1 TO 4 :: DISPLAY AT(
I+13,11):SEG$(A4$,I*3-2,3):: NEX
T I
1670 CALL SOUND(10,1000,4)
1680 FOR I=1 TO 5 :: DISPLAY AT(
I+12,16):SEG$(A5$,I*3-2,3):: NEX
T I
1690 CALL SOUND(10,1000,4)
1700 FOR I=1 TO 5 :: DISPLAY AT(
I+12,21):SEG$(A6$,I*3-2,3):: NEX
T I
1710 CALL SOUND(10,1000,4)
1720 CALL RITARDO(1000):: CALL C
LEAR :: CALL CHARSET
1730 SUBEND
1740 SUB RITARDO(RIT)
1750 FOR I=1 TO RIT :: NEXT I
1760 SUBEND
1770 SUB LINO(R,C,RP)
1780 CALL CHAR(129,"FF")
1790 CALL HCHAR(R,C,129,RP):: CA
LL COLOR(13,7,1)
1800 SUBEND
1810 SUB LINV(R,C,RP)
1820 CALL CHAR(128,"404040404040
4040"):: CALL COLOR(13,7,1)
1830 CALL VCHAR(R,C,128,RP)
1840 SUBEND
1850 SUB TITF
1860 DISPLAY AT(2,2):"IN CASA F.
CASA"
1870 DISPLAY AT(2,20):"1"&"  "&"
2"&"  "&"3"
1880 DISPLAY AT(19,2):"1=schedin
a secondo logica"
1890 DISPLAY AT(20,2):"2=schedin
a casuale"
1900 DISPLAY AT(21,2):"3=schedin
a mista"
1910 DISPLAY AT(23,2):"premi un
tasto per finire"
1920 SUBEND
1930 SUB NUM
1940 CL=2 :: UN=1
1950 FOR RG=4 TO 12 STEP 2 :: DI
SPLAY AT(RG,2)SIZE(2):UN :: UN=U
N+2 :: NEXT RG
1960 FOR RG=14 TO 16 STEP 2 :: D
ISPLAY AT(RG,1)SIZE(3):UN :: UN=
UN+2 :: NEXT RG
1970 SUBEND
```

10-90 - Prima pagina (intestazione).
100 - Richiamo programma (titolo).
120-140 - Inizializzazione.
160-260 - Domande preliminari.
320-450 - Loop per le squadre di casa (A, B, C1).
480–570 - Loop per le squadre di casa (C2).
590-710 - Loop per le squadre in trasferta (A, B, C1).
730-800 - Loop per le squadre in trasferta (C2).
850-980 - Ultima pagina con i risultati.
1000-1210 - Elaborazione dati con verifica max di doppie.
1220-1340 - Parte random.
1360-1470 - Parte mista (logica + random).
1490-1730 - Subroutine titolo principale.
1740-1760 - Subroutine ritardo.
1770-1840 - Subroutine linee orizzontali e verticali.
1850-1920 - Subroutine intestazioni ultima pagina.
1930-1970 - Subroutine numeri progressivi della II e III pagina.

Stringhe e variabili

A$ (26,1) - Stringa che contiene i nomi delle squadre.
RI$ (13,1) - Stringa che contiene i risultati da stampare.
B$ (13,1) - Stringa che contiene i risultati random da stampare.
RIS$ (Q,1) - Stringa che contiene i risultati.
MED$ (I,1) - Stringa con i risultati della media da stampare.
RI2 - Somma dati della C2.
RI, RI1 - Somma dati per (A, B, C1)
SC1, SC2 - Contengono i dati da confrontare.
TT, TOT - Totale delle squadre che giocano in (A, B, C1)
GIO, GIO1 - Partite giocate.
X, A, K, AA, C, CC, Q - Contatori.
V, P, S, D, V1, P1, S1, VV, DD - Contengono i dati delle squadre.

■



Caccia alla coppia per Apple II, C 64 e C 16

Un divertente aiuto per l'esercizio della memoria

di *Claudio Poma*

La naturale evoluzione della rivoluzione informatica sta facendo sentire i suoi effetti anche sul mondo del software e in particolare dei videogiochi, finora incontrastati dominatori. L'interesse dell'acquirente si va spostando verso applicazioni diverse, forse meno colorate e vivaci, ma senz'altro più disponibili a soluzioni personali. Con i giochi di azione in ribasso e quelli gestionali in stallo (ci riferiamo alle applicazioni casalinghe!) forse più per pregiudizio che per una loro presunta scarsa validità applicativa, i programmi oggi più richiesti sono le avventure, i giochi con più partecipanti (chiamarli "di società" mi sembra quantomeno limitativo) e i didattico-educativi.

Il successo di questi, al di là delle differenze che li contraddistinguono, può essere ricercato in un punto comune alle diverse categorie (da non considerare in modo poi tanto "categorico", dato che una avventura può essere costruita in modo da aiutare l'apprendimento di ben definiti argomenti, mentre un programma didattico può essere "giocato" da più persone contemporaneamente) rappresentato dall'interattività: non più solo joystick e numeri, ma la possibilità di poter intervenire attivamente nel modo a noi più consono (lo scambio di messaggi nel linguaggio che utilizziamo tutti i giorni) e, nei migliori prodotti, in una forma molto personalizzata.



L'argomento del programma

Dopo questa breve introduzione passiamo a quanto presentato in queste pagine.
Il programma, in varie versioni per diverse macchine, dopo una breve introduzione con accompagnamento musicale sul ritornello di Yellow Submarine chiede i nomi dei partecipanti (da 1 a 8, ma una minima modifica alle variabili permette il superamento di tale limite) e quindi propone la schermata centrale sulla quale si svolgerà tutta la parte centrale del gioco.
Questa fase è caratterizzata da una griglia di 7 per 6 quadrati di lato, e nella zona dello schermo libera ci sono i nomi dei partecipanti con i relativi punteggi.
Il lampeggio del nome, o l'inverso nel C 64, indica il turno delle giocate, mentre un piccolo cursore nell'angolo in alto a sinistra del primo quadrato della griglia aspetta di essere mosso in sù e giù e a destra e sinistra con i tasti delle freccine.

Per vedere cosa c'è nascosto in un particolare quadrato ci si deve portare con il cursore e quindi premere il tasto Return: sen due soli tentativi si riesce a scoprire una coppia di lettere dell'alfabeto si riceve un punteggio accompagnato da una musichetta di sottofondo e si può continuare con un nuovo tentativo mentre la coppia scoperta rimane fissata sullo schermo, altrimenti un altro nome lampeggia e la mano passa al giocatore successivo mentre i due caratteri appena scoperti scompaiono all'annullamento dei loro pixel.

Scoperte tutte le 21 coppie a disposizione, Yellow Submarine annuncia l'imminente classifica finale a conclusione di un impegno meno risibile di quel che può sembrare da queste righe.

Il listato

Nonostante alcune profonde differenze tra le macchine presentate, i listati sono identici per quanto riguarda la numerazione delle linee e il relativo significato (alcune lievi differenze in questo senso sono dovute al limite di 80 caratteri per linea dei Commodore contro i 255 dell'Apple) al fine di permettere un confronto comparativo fra gli stessi da parte di chi fosse interessato alla conversione dei programmi da una marca all'altra.
Questo lavoro permette anche di trarre alcune considerazioni circa l'incompatibilità tra i vecchi e i nuovi modelli della Commodore.

La differenza esiste, non ci sono dubbi, ma pur essendo sostanziale non è totale bensì a scompartimenti stagni: la grafica e il suono ricorrono a soluzioni completamente diverse e l'unico rimedio è lo sviluppo di una



Caccia alla coppia per Apple II, C 64 e C 16

Listato 1. *Tavola delle forme per Apple. Questi dati sono necessari per la definizione delle lettere in corsivo. Accedere al Monitor battendo Call-151, quindi battere la presente tabella mettendo al posto del meno (–) i due punti (:). Terminata la copiatura tornare al BASIC premendo Ctrl-C e Return e, dopo aver preparato il dischetto dove verrà salvato anche il programma in BASIC* Caccia alla coppia, *digitare Bsave Alfabeto, A36864,L456.*

```
*
9000- 15 00 2C 00 3C 00 56 00
9008- 61 00 77 00 87 00 A0 00
9010- BA 00 D4 00 E1 00 F8 00
9018- 18 01 2E 01 3D 01 60 01
9020- 73 01 81 01 8E 01 9A 01
9028- A9 01 B5 01 1E E7 24 24
9030- 1C 1C 3F F7 1E 36 76 0E
9038- 2D 65 0C 00 0C 0C 0C 0C
9040- 0C 0C 0C 0C 24 E4 3F 1E
9048- 36 36 36 36 36 36 76 2D
9050- 0C 24 24 2C 2D 00 F0 3F
9058- 1F 0C 24 24 64 2D 0E 0E
9060- 00 1C 1C 3F F7 1E 36 76
9068- 0E 2D 65 75 65 DC 23 24
9070- 24 24 24 24 24 04 00 0C
9078- 65 0C 0C 0C 24 1C F7 1E
9080- 36 76 0E 0E 65 0C 00 0C
9088- 0C 0C 0C 0C 0C 24 1C F7
9090- 36 36 36 36 36 36 F6 E7
9098- 24 0C 0C 0C 0C 0C 0C 00
90A0- 1E 1E 1E F7 1E 36 76 65
90A8- 0C 24 24 24 24 24 E4 1C
90B0- 3F 1E 1E 36 76 0E 2D 0C
90B8- 0C 00 0C 0C 0C 0C 0C 0C
90C0- 0C 24 E4 3F 1E 36 36 36
90C8- 36 36 36 36 4D 21 24 24
90D0- 1C F7 1E 00 16 36 F6 1E
90D8- F6 0D 40 30 76 2D 0C 0C
90E0- 00 0C 0C 0C 0C 0C 0C 0C
90E8- 24 E4 3F 1E 36 36 36 36
90F0- 36 36 76 2D 0C 0C 0C 00
90F8- 64 75 36 36 36 05 80 80
9100- 80 80 80 80 28 0E 36 36
9108- 36 05 80 80 80 80 80 80
9110- 28 0E 36 36 36 65 0C 00
9118- 22 0C 2D 0E 36 36 36 05
9120- 80 80 80 80 80 80 28 75
9128- 36 36 76 65 0C 00 F2 36
9130- 36 1E 1E 3F 1C 1C 24 64
9138- 0C 2D 0E 0E 00 0C 0C 0C
9140- 0C 0C 36 36 36 36 36 36
9148- 36 2E 80 80 80 80 80 80
9150- 80 80 80 80 80 80 80 80
9158- 0C 75 36 36 0E 65 0C 00
9160- 1E 1E 3F E7 1C 24 64 0C
9168- 2D 75 0E 36 36 36 36 36
9170- 36 36 00 0C 2D 2D 35 1E
9178- 1E 1E 1E 76 2D 0C 0C 0C
9180- 00 62 0C 0C 0E 0E 0E 36
9188- F6 1E 3F E7 1C 00 3F 27
9190- 34 36 36 36 36 36 0E 65
9198- 0C 00 36 36 36 0E 2D 65
91A0- 24 24 24 95 92 12 65 0C
91A8- 00 64 75 36 36 76 65 24
91B0- 24 24 2D 05 00 2D 2D 2D
91B8- F5 1E 1E 3F 0E 2D DE F3
91C0- 1E 2D 2D 2D 05 00 00 00
```

nuova versione, ma il BASIC essenziale e la gestione dello schermo sono immutati (a parte il Kbyte utilizzato allo scopo: a partire dal 1.024 per il C 64 e dal 3.072 per il C 16), per cui, se non è possibile scambiare i prodotti già preconfezionati, è invece abbastanza semplice adattare i programmi listabili: se si confrontano le versioni qui proposte per il C 16 e il C 64 senza sprite si può notare come le uniche linee riportanti dati diversi siano quelle legate alla routine musicale e poco più.

Alcune note sulle diverse versioni

Per Apple si è fatto ricorso alla pagina grafica in alta risoluzione che ha permesso la definizione dei caratteri in corsivo. Questi sono memorizzati nel listato Tavola delle forme (listato 1) e per salvarli su dischetto è necessario accedere al Monitor (Call-151) e quindi, iniziando a digitare 9000: 15 00 2C ecc. si devono immettere tutti i valori, nessuno escluso, pena la mancata definizione dei caratteri desiderati. Terminata l'operazione e controllati eventuali errori si deve tornare in BASIC (Ctrl-C e Return) per battere:

BSAVE ALFABETO,A36864,L456 (decimali)

o:

BSAVE ALFABETO,A$9000,L$1C7 (esadecimali con $)

avendo cura di aver predisposto il dischetto sul quale si intende memorizzare anche il programma in BASIC del listato 2. Questo è importante in quanto altrimenti la linea 40 darebbe un risultato nullo, bloccando, di conseguenza, l'esecuzione del programma.

E' stata fatta la scelta della routine separata dal corpo del programma per permettere l'utilizzo di tali figure in qualsiasi altro programma senza richiedere la trascrizione di tutti i dati necessari.

Chi non gradisse questa soluzione dovrà tradurre tutti i numeri esadecimali nei rispettivi decimali e metterli in linee Data da leggere oppor-

PERSONAL SOFTWARE

Caccia alla coppia per Apple II, C 64 e C 16

Listato 2. *Caccia alla coppia, versione Apple.*

```
4   HCOLOR= 3
5   DIM FO(6,5),CA(21),O(21),P(21),
     KJ(21)
6   READ A$: IF A$ <  > "+" THEN 6
7   FOR T = 770 TO 792: READ A: POKE
     T,A: NEXT
8   FOR T = 1 TO 21:KJ(T) = 1: NEXT

9 UU(1) = 2:UU(2) = 1.4
10   HOME : SCALE= 1
20   HIMEM: 36860
25   IF GQ = 1 THEN 45
30   POKE 232,0: POKE 233,144
40   PRINT  CHR$ (4)"BLOAD ALFABETO
      ,A36864"
45 GQ = 0: RESTORE
50   READ A$: IF A$ <  > "*" THEN 5
     0
60   FOR T = 1 TO 21: READ O(T),P(T
     ): NEXT
70   FOR T = 1 TO 8:PU(T) = 0:RE(T)
      =  - 1:RE$(T) = "": NEXT
80 KK = 0
100 X = 10:Y = 40: HGR
110  FOR T = 1 TO 21
115 X = X + 20: IF X > 235 THEN X =
      40:Y = 90
120  DRAW T AT X + O(T),Y + P(T)
130  NEXT
140  HPLOT 10,30 TO 270,30 TO 270,
      120 TO 10,120 TO 10,30
145  FOR T = 1 TO 21:CA(T) = 0: NEXT

147  GOSUB 1400
150  FOR T = 0 TO 6: FOR R = 0 TO
      5
160 CA =  INT ( RND (1) * 21 + 1):
      CA(CA) = CA(CA) + 1: IF CA(CA
      ) > 2 THEN 160
170 FO(T,R) = CA: NEXT R,T
200  VTAB 21: INPUT "IN QUANTI GIO
      CHIAMO ? ";GI
210  IF GI < 1 OR GI > 8 THEN  CALL
      - 912: GOTO 200
220  FOR T = 1 TO GI
230  PRINT "Nome del giocatore N."
      T;: INPUT "  ";GI$(T)
240  IF  LEN (GI$(T)) > 12 THEN  PRINT
      "Nome troppo lungo, Max.12 ":
       GOTO 230
250  NEXT
500  HOME : HGR : HCOLOR= 3
510  FOR X = 50 TO 225 STEP 25
520  HPLOT X,5 TO X,155
530  NEXT
540  FOR Y = 5 TO 155 STEP 25
550  HPLOT 50,Y TO 225,Y
560  NEXT
600 V = 21:H = 1
610  FOR T = 1 TO GI
620 V(T) = V:H(T) = H
625  VTAB V(T): HTAB H(T): PRINT G
     I$(T);: HTAB (H(T) + 16 -  LEN
     ( STR$ (PU(T)))): PRINT PU(T)
     ;
630 H = H + 20: IF H > 21 THEN H =
     1:V = V + 1
640  NEXT
650 CA =  INT ( RND (1) * GI + 1)
700  VTAB V(CA): HTAB H(CA): FLASH
     : PRINT GI$(CA);
705  NORMAL
710 X = 0:Y = 0:A = 0:B = 0
720 Q = 52 + X * 25:W = 7 + 25 * Y

730  HPLOT Q,W TO Q + 3,W TO Q + 3
     ,W + 3 TO Q,W + 3 TO Q,W
734  POKE  - 16368,0
735  FOR T = 1 TO 200: NEXT
740 RR =  PEEK ( - 16384) - 128: IF
     RR = 13 THEN 900
750  IF RR = 11 AND Y > 0 THEN B =
     - 1: GOTO 790
760  IF RR = 10 AND Y < 5 THEN B =
     1: GOTO 790
770  IF RR = 8 AND X > 0 THEN A =
     - 1: GOTO 790
780  IF RR = 21 AND X < 6 THEN A =
     1
790  HCOLOR= 0: HPLOT Q,W TO Q + 3
     ,W TO Q + 3,W + 3 TO Q,W + 3 TO
     Q,W: HCOLOR= 3
800 X = X + A:Y = Y + B:A = 0:B =
     0
810  GOTO 720
900  IF FI = 1 AND (C = X AND D =
     Y) THEN 750
905  IF FO(X,Y) = 0 THEN 750
910 FI = FI + 1:FI(FI) = FO(X,Y):C
     (FI) = Q:D(FI) = W: IF FI = 1
      THEN C = X:D = Y
913 KJ(FI(FI)) = KJ(FI(FI)) + .05
914 TY(4 + FI) = TY(2 + FY):TY(2 +
     FI) = TY(FI):TY(FI) = FI(FI)
915 C(FI) = C(FI) + O(FI(FI)):D(FI
     ) = D(FI) + P(FI(FI))
920  DRAW FI(FI) AT C(FI),D(FI)
930  IF FI < 2 THEN 750
950  IF FI(1) <  > FI(2) THEN  FOR
     T = 1 TO 3000: NEXT : GOTO 11
     00
960  GOTO 1040
970  FOR T = 0 TO 6: FOR R = 0 TO
     5
980  IF FO(T,R) = FI(1) THEN FO(T,
     R) = 0
990  NEXT R,T
1000 PU =  INT ((41 - KK) / KJ(FI(
     FI)))
1010  FOR T = 3 TO 6: IF TY(T) = F
     I(FI) THEN PU =  INT (PU / UU
     ( INT (T / 2.5)) + 1)
1015  NEXT
1020 PU(CA) = PU(CA) + PU
1030  VTAB V(CA): HTAB (H(CA) + 16
     -  LEN ( STR$ (PU(CA)))): PRINT
     PU(CA);
1035  GOTO 1090
1040  RESTORE
1050  READ A$: IF A$ <  > "A" THEN
     1050
```

Caccia alla coppia per Apple II, C 64 e C 16

tunamente come spiegato nell'articolo *Grafica e Joystick per Apple* apparso sul n. 24 di **Personal Software**. Per il C 16 (listato 3), non è preso in considerazione il ricorso all'alta risoluzione a causa dei problemi che questo comporta nei confronti della memoria: infatti dei 12 Kbyte gestibili dall'utente, ben 10 vengono "mangiati" dall'Hires per cui non rimane abbastanza spazio per l'organizzazione di programmi sufficientemente complessi. Ma ci ritorneremo. La versione per C 64 è invece presente nel listato 4.

Come si può facilmente vedere, mancano le istruzioni per il gioco. Non si tratta di una distrazione, ma di una scelta dovuta alla presentazione di diversi listati per non aumentare lo spazio occupato con lunghe liste di Print identiche. Per chi avesse delle difficoltà nello sviluppo personale di questa operazione, ecco un esempio valido per tutte le versioni (basta aggiungere queste linee al listato della propria macchina):

- 1 Gosub 2500
- 2500 Rem Istruzioni
- 2510 ? "Il gioco consiste nello scoprire le coppie di lettere";
- 2515 ? "nascoste dietro una griglia".
- 2520 ? "Il giocatore di turno è indicato con il proprio nome in rilievo".
- 2530 ? "Per muovere il piccolo cursore lampeggiante usare i tasti con le frecce".
- 2540 ? "Per vedere cosa c'è nel riquadro occupato dal cursore premere Return".
- 2550 ? "Nel caso si scopra una coppia si riceverà un punteggio e si terrà la mano";
- 2560 ? "altrimenti il gioco passa al giocatore successivo".
- 2570 ? "Alla fine verrà stilata una classifica in relazione ai punteggi conseguiti".
- 2580 ?:? "Premi Return per continuare".
- 2590 InputA$:Return.

Linea per linea

Fino alla linea 25 si hanno il di-

Seguito Caccia alla coppia per Apple.

```
1060  FOR T = 1 TO 6: READ A1,A2: POKE
      768,A1: POKE 769,A2: CALL 770
      : NEXT
1070  GOTO 970
1090 KK = KK + 1: HCOLOR= 0: GOTO
      1105
1100  HCOLOR= 0: DRAW FI(1) AT C(1
      ),D(1): DRAW FI(2) AT C(2),D(
      2)
1105  HPLOT Q,W TO Q + 3,W TO Q +
      3,W + 3 TO Q,W + 3 TO Q,W
1110  VTAB V(CA): HTAB H(CA): PRINT
      GI$(CA);
1120  HCOLOR= 3: IF FI(1) = FI(2) THEN
      1140
1130 CA = CA + 1: IF CA > GI THEN
      CA = 1
1140 FI = 0:PU = 0
1150  IF KK = 21 THEN 1190
1160  GOTO 700
1190  FOR T = 1 TO GI:RE(GI) =  -
      1: NEXT
1200  FOR T = 1 TO GI
1210  FOR T1 = 1 TO GI - 1
1220  IF PU(T) > RE(T1) THEN 1250
1230  NEXT T1
1240  GOTO 1270
1250  FOR T2 = GI - 1 TO T1 STEP  -
      1
1255 RE(T2 + 1) = RE(T2):RE$(T2 +
      1) = RE$(T2)
1260  NEXT T2
1270 RE(T1) = PU(T):RE$(T1) = GI$(
      T)
1280  NEXT T
1290  HOME : TEXT
1300  VTAB 10 - GI: HTAB 12: FLASH
      : PRINT "CLASSIFICA FINALE": NORMAL
1310 K = 12 + GI: GOSUB 1400
1320  FOR T = GI TO 1 STEP  - 1
1330  VTAB K: HTAB 11: PRINT T")
      "RE$(T);: HTAB (30 -  LEN ( STR$
      (RE(T)))): PRINT RE(T)
1340 K = K - 2: FOR T1 = 1 TO 500:
      NEXT
1350  NEXT
1360  VTAB 24: PRINT "Volete gioca
      re ancora ? (S/N) ";
1370  GET A$: IF A$ = "N" THEN  HOME
      : END
1380  IF A$ = "S" THEN GQ = 1: GOTO
      8
1390  GOTO 1370
1400  RESTORE
1410  READ A$: IF A$ <  > "A" THEN
      1410
1420  READ A1,A2: IF A1 = 0 THEN 1
      450
1430  POKE 768,A1: POKE 769,A2: CALL
      770
1440  GOTO 1420
1450  RETURN
2000  DATA *,18,15,8,17,14,15,14,1
      4,9,17,9,15,19,9,8,17,12,7,8,
      17,8,12,7,12,16,7,7,11,15,11,
      9,11,9,11,14,9,10,9,7,11,9,8
2010  DATA +,173,48,192,136,208,5,
      206,1,3,240,9,202,208,245,174
      ,0,3,76,2,3,96,0,0
2020  DATA A,102,180,86,255,102,64
      ,115,128,102,64,128,255,102,1
      28,102,64,115,128,128,64,152,
      255,152,64,102,128,102,64,115
      ,255
2030  DATA 0,0
```

Caccia alla coppia per Apple II, C 64 e C 16

Listato 3. *Caccia alla coppia, versione C 16.*

```
5 DIMFO(6,5),CA(21),KJ(26),W(21)
6 RI$="-{ 2 SPAZI}-{ 2 SPAZI}-{ 2 SPAZI}
  -{ 2 SPAZI}-{ 2 SPAZI}-{ 2 SPAZI}-
  T 2 SPAZI}-":R2$="[<Q>]{ 2 *}+{ 2 *}+
  { 2 *}+{ 2 *}+{ 2 *}+{ 2 *}+{ 2 *}[<W>]
  "
8 FORT=1TO26:KJ(T)=1:NEXT
9 UU(1)=2:UU(2)=1.4
10 PRINT"{CLR}"
25 IFGQ=1THEN45
29 RESTORE
30 READA$:IFA$<>"L"THEN30
40 FORT=1TO21:READW(T):NEXT
45 GQ=0:RESTORE
70 FORT=1TO8:PU(T)=0:RE(T)=-1:RE$(T)="":N
   EXT
80 KK=0
100 PRINT"{CLR}{ 5 GIU'}"CHR$(14)
110 PRINTTAB(2)"[<A>]{ 33 *}[<S>]":PRINTT
    AB(2)"-"SPC(33)"-"
120 PRINTTAB(2)"-{ 3 SPAZI}A{ 2 SPAZI}B
    { 2 SPAZI}C{ 2 SPAZI}D{ 2 SPAZI}E
    { 2 SPAZI}F{ 2 SPAZI}G{ 2 SPAZI}H
    { 2 SPAZI}I{ 2 SPAZI}L{ 2 SPAZI}-"
130 PRINTTAB(2)"-"SPC(33)"-":PRINTTAB(2)"
    - M{ 2 SPAZI}N{ 2 SPAZI}O{ 2 SPAZI}P
    { 2 SPAZI}Q{ 2 SPAZI}R{ 2 SPAZI}S
    { 2 SPAZI}T{ 2 SPAZI}U{ 2 SPAZI}V
    { 2 SPAZI}Z -"
140 PRINT"{ 2 SPAZI}-"SPC(33)"-":PRINT"
    { 2 SPAZI}[<Z>]{ 33 *}[<X>]"
145 FORT=1TO21:CA(T)=0:NEXT
147 GOSUB1400
150 FORT=0TO6:FORR=0TO5
160 CA=INT(RND(1)*21+1):CA(CA)=CA(CA)+1:I
    FCA(CA)>2THEN160
170 FO(T,R)=W(CA):NEXTR,T
200 PRINT:PRINT:INPUT"IN QUANTI GIOCHIAMO
    ";GI
210 IFGI<1ORG>8THENPRINT"{ 2 SU}":GOTO200

220 FORT=1TOGI
230 PRINT"NOME DEL GIOCATORE N."T;:INPUTG
    I$(T)
240 IFLEN(GI$(T))>12THENPRINT"NOME TROPPO
     LUNGO, MAX.12 ":GOTO230
250 NEXT
500 PRINT"{CLR}"
510 PRINTTAB(18)"[<A>]{ 2 *}[<R>]{ 2 *}
    [<R>]{ 2 *}[<R>]{ 2 *}[<R>]{ 2 *}
    [<R>]{ 2 *}[<R>]{ 2 *}[<S>]"
520 FORT=1TO5:FORK=1TO2:PRINTTAB(18)RI$:N
    EXT:PRINTTAB(18)R2$:NEXT
530 FORT=1TO2:PRINTTAB(18)RI$:NEXT
540 PRINTTAB(18)"[<Z>]{ 2 *}[<E>]{ 2 *}
    [<E>]{ 2 *}[<E>]{ 2 *}[<E>]{ 2 *}
    [<E>]{ 2 *}[<E>]{ 2 *}[<X>]"
600 PRINT"{HOME}"
610 FORT=1TOGI
620 PRINT:PRINTGI$(T);SPC(16-LEN(STR$(PU(
    T)))-LEN(GI$(T)));PU(T)
640 NEXT
650 CA=INT(RND(1)*GI+1)
700 Z$="{HOME}":FORT=1TOCA*2:Z$=Z$+"
    {GIU'}":NEXT:PRINTZ$;"{RVS}{FLASH ON}
    ";GI$(CA);"{FLASH OF}{OFF}"
710 X=0:Y=0:A=0:B=0
720 POKE3171+X*3+Y*40*3,126
735 FORT=1TO50:NEXT
740 GETA$:IFA$=CHR$(13)THEN900
750 IFA$="{SU}"ANDY>0THENB=-1:GOTO790
760 IFA$="{GIU'}"ANDY<5THENB=1:GOTO790
770 IFA$="{SIN}"ANDX>0THENA=-1:GOTO790
780 IFA$="{DES}"ANDX<6THENA=1
790 POKE3171+X*3+Y*40*3,32
800 X=X+A:Y=Y+B:A=0:B=0
810 GOTO720
900 IFFI=1AND(C=XANDD=Y)THEN750
905 IFFO(X,Y)=0THEN750
910 FI=FI+1:FI(FI)=FO(X,Y):C(FI)=X:D(FI)=
    Y:IFFI=1THENC=X:D=Y
913 KJ(FI(FI))=KJ(FI(FI))+.05
914 TY(4+FI)=TY(2+FI):TY(2+FI)=TY(FI):TY(
    FI)=FI(FI)
920 POKE3212+C(FI)*3+D(FI)*40*3,FI(FI)
930 IFFI<2THEN750
950 IFFI(1)<>FI(2)THENFORT=1TO3000:NEXT:G
    OTO1100
960 GOTO1040
970 FORT=0TO6:FORR=0TO5
980 IFFO(T,R)=FI(1)THENFO(T,R)=0
990 NEXTR,T
1000 PU=INT((41-KK)/KJ(FI(FI)))
1010 FORT=3TO6:IFTY(T)=FI(FI)THENPU=INT(P
     U/UU(INT(T/2.5))+1)
1015 NEXT
1020 PU(CA)=PU(CA)+PU
1030 J$="{HOME}":FORT=1TOCA*2:J$=J$+"
     {GIU'}":NEXT:PRINTJ$;SPC(16-LEN(STR$
     (PU(CA))))PU(CA)
1035 GOTO1090
1040 RESTORE
1050 A4=911:GOSUB1410
1070 GOTO970
1090 KK=KK+1:GOTO1105
1100 POKE3212+C(1)*3+D(1)*3*40,32:POKE321
     2+C(2)*3+D(2)*3*40,32
1105 POKE3171+X*3+Y*40*3,32
1110 PRINTZ$;GI$(CA)
1120 IFFI(1)=FI(2)THEN1140
1130 CA=CA+1:IFCA>GITHENCA=1
1140 FI=0:PU=0
1150 IFKK=21THEN1190
1160 GOTO700
1190 FORT=1TOGI:RE(GI)=-1:NEXT
1200 FORT=1TOGI
1210 FORT1=1TOGI-1
1220 IFPU(T)>RE(T1)THEN1250
1230 NEXTT1
1240 GOTO1270
1250 FORT2=GI-1TOT1STEP-1
1255 RE(T2+1)=RE(T2):RE$(T2+1)=RE$(T2)
1260 NEXTT2
1270 RE(T1)=PU(T):RE$(T1)=GI$(T)
1280 NEXTT
1290 PRINT"{CLR}"
1300 Z$="{HOME}":FORT=1TO11-GI:Z$=Z$+"
     {GIU'}":NEXT:PRINTZ$;TAB(12)"{RVS}
     {FLASH ON}CLA{ 2 S}IFICA FINALE
     {OFF}"
```

PERSONAL SOFTWARE

Caccia alla coppia per Apple II, C 64 e C 16

Seguito Caccia alla coppia per C 16.

```
1310 K=11+GI:GOSUB1400
1320 FORT=GITO1STEP-1
1330 Z$="{HOME}":FORU=1TOK:Z$=Z$+"{GIU'}"
     :NEXT
1335 PRINTZ$;TAB(9)T"){ 2 SPAZI}"RE$(T);S
     PC(15-LEN(RE$(T))-LEN(STR$(RE(T))))R
     E(T)
1340 K=K-2:FORT1=1TO500:NEXT
1350 NEXT
1360 PRINT"{HOME}":FORT=1TO22:PRINT:NEXT:
     PRINT"VOLETE GIOCARE ANCORA ? (S/N)
     ";
1370 GETA$:IFA$="N"THENPRINT"{CLR}":END
1380 IFA$="S"THENGQ=1:GOTO8
1390 GOTO1370
1400 RESTORE:VOL8:A4=0
1410 READA$:IFA$<>"A"THEN1410
1420 READA1,A2:IFA1=A4ANDA2=20THEN1450
1430 SOUND2,A1,A2
1440 GOTO1420
1450 RETURN
2000 DATAL,1,2,3,4,5,6,7,8,9,12,13,14,15,
     16,17,18,19,20,21,22,26
2020 DATAA,911,35,929,40,911,10,897,20,91
     1,10,881,40,911,20,911,10,897,20
2025 DATA881,10,854,40,854,10,911,20,911,
     10,897,40,0,20
```

Listato 4. *Caccia alla coppia, versione C 64.*

```
5 DIMFO(6,5),CA(21),KJ(26),W(21)
6 RI$="-{ 2 SPAZI}-{ 2 SPAZI}-{ 2 SPAZI}
  -{ 2 SPAZI}-{ 2 SPAZI}-{ 2 SPAZI}-
  [ 2 SPAZI}-":R2$="[<Q>]{ 2 *}+{ 2 *}+
  { 2 *}+{ 2 *}+{ 2 *}+{ 2 *}+{ 2 *}[<W>]
  "
8 FORT=1TO26:KJ(T)=1:NEXT
9 UU(1)=2:UU(2)=1.4
10 PRINT"{CLR}"
25 IFGQ=1THEN45
29 RESTORE
30 READA$:IFA$<>"L"THEN30
40 FORT=1TO21:READW(T):NEXT
45 GQ=0:RESTORE
70 FORT=1TO8:PU(T)=0:RE(T)=-1:RE$(T)="":N
   EXT
80 KK=0
100 PRINT"{CLR}{ 5 GIU'}"CHR$(14)
110 PRINTTAB(2)"[<A>]{ 33 *}[<S>]":PRINTT
    AB(2)"-"SPC(33)"-"
120 PRINTTAB(2)"-{ 3 SPAZI}A{ 2 SPAZI}B
    { 2 SPAZI}C{ 2 SPAZI}D{ 2 SPAZI}E
    { 2 SPAZI}F{ 2 SPAZI}G{ 2 SPAZI}H
    { 2 SPAZI}I{ 2 SPAZI}L{ 2 SPAZI}-"
130 PRINTTAB(2)"-"SPC(33)"-":PRINTTAB(2)"
    - M{ 2 SPAZI}N{ 2 SPAZI}O{ 2 SPAZI}P
    { 2 SPAZI}Q{ 2 SPAZI}R{ 2 SPAZI}S
    { 2 SPAZI}T{ 2 SPAZI}U{ 2 SPAZI}V
    { 2 SPAZI}Z -"
140 PRINT"{ 2 SPAZI}-"SPC(33)"-":PRINT"
    { 2 SPAZI}[<Z>]{ 33 *}[<X>]"
145 FORT=1TO21:CA(T)=0:NEXT
147 GOSUB1400
150 FORT=0TO6:FORR=0TO5
160 CA=INT(RND(1)*21+1):CA(CA)=CA(CA)+1:I
    FCA(CA)>2THEN160
170 FO(T,R)=W(CA):NEXTR,T
200 PRINT:PRINT:INPUT"IN QUANTI GIOCHIAMO
    ";GI
210 IFGI<1ORG>8THENPRINT"{ 2 SU}":GOTO200
220 FORT=1TOGI
230 PRINT"NOME DEL GIOCATORE N."T;:INPUTG
    I$(T)
240 IFLEN(GI$(T))>12THENPRINT"NOME TROPPO
     LUNGO, MAX.12 ":GOTO230
250 NEXT
500 PRINT"{CLR}"
510 PRINTTAB(17)"[<A>]{ 2 *}[<R>]{ 2 *}
    [<R>]{ 2 *}[<R>]{ 2 *}[<R>]{ 2 *}
    [<R>]{ 2 *}[<R>]{ 2 *}[<S>]"
520 FORT=1TO5:FORK=1TO2:PRINTTAB(17)RI$:N
    EXT:PRINTTAB(17)R2$:NEXT
530 FORT=1TO2:PRINTTAB(17)RI$:NEXT
540 PRINTTAB(17)"[<Z>]{ 2 *}[<E>]{ 2 *}
    [<E>]{ 2 *}[<E>]{ 2 *}[<E>]{ 2 *}
    [<E>]{ 2 *}[<E>]{ 2 *}[<X>]"
600 PRINT"{HOME}"
610 FORT=1TOGI
620 PRINT:PRINTGI$(T);SPC(16-LEN(STR$(PU(
    T))-LEN(GI$(T)));PU(T)
640 NEXT
650 CA=INT(RND(1)*GI+1)
700 Z$="{HOME}":FORT=1TOCA*2:Z$=Z$+"
    {GIU'}":NEXT:PRINTZ$;"{RVS}";GI$(CA);
    "{OFF}"
710 X=0:Y=0:A=0:B=0
720 POKE1122+X*3+Y*40*3,126
735 FORT=1TO50:NEXT
740 GETA$:IFA$=CHR$(13)THEN900
750 IFA$="{SU}"ANDY>0THENB=-1:GOTO790
760 IFA$="{GIU'}"ANDY<5THENB=1:GOTO790
770 IFA$="{SIN}"ANDX>0THENA=-1:GOTO790
780 IFA$="{DES}"ANDX<6THENA=1
790 POKE1122+X*3+Y*40*3,32
800 X=X+A:Y=Y+B:A=0:B=0
810 GOTO720
900 IFFI=1AND(C=XANDD=Y)THEN750
905 IFFO(X,Y)=0THEN750
910 FI=FI+1:FI(FI)=FO(X,Y):C(FI)=X:D(FI)=
    Y:IFFI=1THENC=X:D=Y
913 KJ(FI(FI))=KJ(FI(FI))+.05
914 TY(4+FI)=TY(2+FI):TY(2+FI)=TY(FI):TY(
    FI)=FI(FI)
920 POKE1163+C(FI)*3+D(FI)*40*3,FI(FI)
930 IFFI<2THEN750
950 IFFI(1)<>FI(2)THENFORT=1TO3000:NEXT:G
    OTO1100
960 GOTO1040
970 FORT=0TO6:FORR=0TO5
980 IFFO(T,R)=FI(1)THENFO(T,R)=0
990 NEXTR,T
800 PU=INT((41-KK)/KJ(FI(FI)))
1010 FORT=3TO6:IFTY(T)=FI(FI)THENPU=INT(P
     U/UU(INT(T/2.5))+1)
1015 NEXT
1020 PU(CA)=PU(CA)+PU
1030 J$="{HOME}":FORT=1TOCA*2:J$=J$+"
     {GIU'}":NEXT:PRINTJ$;SPC(16-LEN(STR$
     (PU(CA))))PU(CA)
1035 GOTO1090
```



Caccia alla coppia per Apple II, C 64 e C 16

Seguito Caccia alla coppia per C 64.

```
1040 RESTORE
1050 A4=65:GOSUB1410
1070 GOTO970
1090 KK=KK+1:GOTO1105
1100 POKE1163+C(1)*3+D(1)*3*40,32:POKE116
     3+C(2)*3+D(2)*3*40,32
1105 POKE1122+X*3+Y*40*3,32
1110 PRINTZ$;GI$(CA)
1120 IFFI(1)=FI(2)THEN1140
1130 CA=CA+1:IFCA>GITHENCA=1
1140 FI=0:PU=0
1150 IFKK=21THEN1190
1160 GOTO700
1190 FORT=1TOGI:RE(GI)=-1:NEXT
1200 FORT=1TOGI
1210 FORT1=1TOGI-1
1220 IFPU(T)>RE(T1)THEN1250
1230 NEXTT1
1240 GOTO1270
1250 FORT2=GI-1TOT1STEP-1
1255 RE(T2+1)=RE(T2):RE$(T2+1)=RE$(T2)
1260 NEXTT2
1270 RE(T1)=PU(T):RE$(T1)=GI$(T)
1280 NEXTT
1290 PRINT"{CLR}"
1300 Z$="{HOME}":FORT=1TO11-GI:Z$=Z$+"
     {GIU'}":NEXT:PRINTTZ$;TAB(12)"{RVS}
     {RVS}CLA{ 2 S}IFICA FINALEE{OFF}"
1310 K=11+GI:GOSUB1400
1320 FORT=GITO1STEP-1
1330 Z$="{HOME}":FORU=1TOK:Z$=Z$+"{GIU'}"
     :NEXT
1335 PRINTZ$;TAB(9)T"){ 2 SPAZI}"RE$(T);S
     PC(15-LEN(RE$(T))-LEN(STR$(RE(T))))R
     E(T)
1340 K=K-2:FORT1=1TO500:NEXT
1350 NEXT
1360 PRINT"{HOME}":FORT=1TO22:PRINT:NEXT:
     PRINT"VOLETE GIOCARE ANCORA ? (S/N)
     ";
1370 GETA$:IFA$="N"THENPRINT"{CLR}":END
1380 IFA$="S"THENGQ=1:GOTO8
1390 GOTO1370
1400 RESTORE:A4=0
1405 FL=54272:FH=FL+1:WW=FL+4:AA=FL+5:HH=
     FL+6:LL=FL+24:A4=0
1410 READA$:IFA$<>"A"THEN1410
1420 READA1,A2,A3:IFA1=A4ANDA3=50THEN1450

1430 POKELL,10:POKEAA,50:POKEHH,68:POKEFH
     ,A1:POKEFL,A2:POKEWW,17
1435 FORT=1TOA3*5:NEXT:POKEWW,0
1440 GOTO1420
1450 RETURN
2000 DATAL,1,2,3,4,5,6,7,8,9,12,13,14,15,
     16,17,18,19,20,21,22,26
2020 DATAA,65,181,85,78,36,100,65,181,25,
     58,138,50,65,181,25,52,39,100
2023 DATA65,181,50,65,181,25,58,138,50,52
     ,39,25,43,219,100,43,219,25
2025 DATA65,181,50,65,181,25,58,138,100,0
     ,0,50
```

mensionamento e la definizione di alcune variabili. L'unica differenza è nella Dim della variabile KJ che varia da 21 nell'Apple a 26 nei Commodore in quanti per questi, facendo ricorso al codice ASCII per la scrittura dei diversi caratteri, è risultato più semplice sovradimensionare la variabile lasciando i vuoti corrispondenti alle lettere X Y K J W non utilizzate nel gioco.

30-40 - Vengono caricati i caratteri utilizzati: tavola delle forme per Apple, codici ASCII per Commodore.

50-60 - (Solo Apple) - Caricano i valori per assestare correttamente le lettere in alta risoluzione all'interno dei quadrati della griglia.

70-80 - Annullamento eventuali punteggi di partite precedenti e numeri di coppie scoperte.

100-140 - Composizione dello schermo iniziale con presentazione dei caratteri. In alta risoluzione per Apple.

145 - Annullamento posizione lettere di una eventuale partita precedente.

147 - Rimanda alla routine sonora per la musichetta introduttiva.

150-170 - Vengono casualmente decise le posizioni nella griglia delle 21 coppie di lettere.

200-250 - Vengono chiesti e controllati il numero e i nomi di chi gioca.

500-540 - (560 solo per Apple) - Questa routine disegna la griglia sulla quale si svolgerà il gioco: in alta risoluzione per Apple, con caratteri semigrafici per C 16 e C 64. La differenza del valore dei Tab tra questi due è dovuta al loro diverso modo di gestire lo schermo: il C 64, quando viene occupata l'ultima colonna della riga con l'ultimo carattere della stringa, va a capo prima della fine della stampa per cui la stessa fa saltare un'ulteriore linea e la stampa successiva sarà data con una riga di spazio, mentre il C 16 nella stessa situazione si comporta in modo diverso e il risultato è l'assenza della riga vuota. Il problema poteva essere risolto lasciando i valori delle Tab invariati e aggiungendo dei punti e virgola (;) finali nella versione per C 64.

600-640 - Vengono posizionati i nomi dei partecipanti con i relativi punteggi. Nell'Apple, per i problemi relativi alla scrittura sulla pagina in alta risoluzione, si è scelta l'opzione Hires + 4 righe di testo e quindi i nomi compaiono ovviamente sul fondo dello schermo, mentre sul C 16 e sul C 64 tali dati sono visualizzati sulla sinistra del video avendo posto la griglia a destra.

650 - Scelta del giocatore a cui tocca iniziare.

700 - Lampeggio del nome del giocatore di turno (inverso per C 64).

710-735 - Annullamento della posizione del cursore di griglia e suo posizionamento con seguente pausa per permetterne la visione.

740-780 - Ciclo di controllo per lo spostamento del cursore con l'uso dei tasti con freccette e per la visione delle lettere nascoste tramite la pressione del tasto Return.

790-810 - Cancellazione del cursore con aggiornamento delle sue variabili e ritorno alla linea per la stampa



Caccia alla coppia per Apple II, C 64 e C 16

5-80	Dimensionamento variabili e preparazione caratteri.
100-140	Schermo iniziale.
150-170	Posizionamento casuale delle lettere nella griglia.
200-250	Numero e nomi dei giocatori.
500-560	Visualizzazione della griglia.
600-640	Posizionamento dei nomi dei giocatori sullo schermo.
700-730	Preparazione per l'inizio del gioco.
740-810	Gestione spostamento cursore e visualizzazione delle lettere.
900-1160	Controllo di quanto trovato con relative conseguenze.
1190-1280	Preparazione della classifica finale.
1290-1390	Stampa della classifica finale.
1400-1450	Routine musicale.
1500-1580	Routine gestione sprite (solo C 64).

Tabella 1. *Elenco delle principali routine.*

nella eventuale nuova posizione. In caso di mancato movimento genera l'effetto di lampeggio.

900 - Annulla l'effetto del tasto Return se ci si trova nella casella appena scoperta.

905 - Annulla l'effetto del tasto Return se ci si trova in una casella contenente il carattere di una coppia già scoperta.

910 - Assegnazione delle diverse variabili in gioco per la nuova casella aperta.

913 - Aumento del coefficiente correlato al numero di volte che una lettera è stata scoperta senza l'ottenimento della coppia.

914 - Assegnazione delle variabili che memorizzano gli ultimi caratteri scoperti al fine del punteggio.

920 - Stampa il carattere contenuto nel riquadro richiesto.

930 - Rimanda allo spostamento del cursore se si è alla prima scelta.

950 - Se le due figure scoperte non sono uguali passa alla cancellazione delle stesse.

970-990 - Vengono annullati i valori dei riquadri corrispondenti alla coppia scoperta per permettere il controllo che la linea 905 effettuerà nelle mani successive.

1000-1020 - Viene assegnato il punteggio in considerazione: del numero di coppie già ritrovate (KK), del coefficiente relativo a precedenti individuazioni delle lettere in questione (KJ linea 913), del tempo passato dai precedenti ritrovamenti (linea 914). Per quest'ultimo caso il punteggio viene dimezzato se uno dei due caratteri della coppia è stato scoperto nel turno precedente, mentre viene ridotto del 40% se è stato scoperto due turni prima.

1030 - Aggiorna il punteggio.

1040-1050 - (1060 per Apple) - Breve musichetta di accompagnamento al ritrovamento.



Caccia alla coppia per Apple II, C 64 e C 16

1090 - Aumenta la variabile utilizzata per il conteggio delle coppie ritrovate.
1100 - Cancella i caratteri se non uguali (da linea 950).
1105 - Cancella il cursore.
1110 - Riporta il nome del giocatore che ha finito in caratteri normali.
1120 - Lascia il turno allo stesso giocatore se questi ha indovinato la coppia.
1130 - Aggiorna il turno dei giocatori se l'attuale ha fallito il compito.
1190-1280 - Questa routine mette in ordine i vari giocatori in relazione al punteggio degli stessi per la visualizzazione della classifica finale.
1290-1390 - Stampa della classifica finale con musica (1310) e richiesta per una eventuale ulteriore partita.
1400-1450 - Diverse routine musicali in relazione alle capacità in tal senso dei computer utilizzati.
2000 - Per Apple valori per la correzione del posizionamento dei caratteri in alta risoluzione. Per C 64 e C 16 valori per codice ASCII dei caratteri.
2010 - (Solo Apple) - Routine in linguaggio macchina per musica.
2020 - E seguenti, diversi valori richiesti dalle varie macchine per l'esecuzione del ritornello di Yellow Submarine.

Elenco principali variabili

FO(6,5) - Memorizza le posizioni delle lettere nella griglia.
CA(21) - Serve per il posizionamento casuale delle lettere.
O(21) P(21) - (Solo per Apple) - Contengono i valori per il corretto posizionamento delle lettere nella griglia.
KJ(21) - Controlla il coefficiente correlato al numero di volte che una lettera è stata scoperta senza l'ottenimento della coppia.
UU(2) - Contiene il coefficiente di riduzione del punteggio per le lettere scoperte nelle due volte precedenti.
GI - Numero giocatori.
GI$(GI) - Nomi dei giocatori.
X, Y - Controllo del cursore di griglia.
A, B - Controllo del movimento del cursore.
TY(6) - Memorizza le ultime lettere estratte per il punteggio.
FI - Numero delle figure visionate nella mano in corso.
FI(FI) - Numero di codice corrispondente alla lettera in visione.
KK - Numero di coppie scoperte.
PU(GI) - Punteggio dei giocatori.
RE(GI) RE$(GI) - Punti e nomi dei giocatori per la classifica finale.

Conclusioni

Una prima occhiata potrebbe farlo considerare uno dei tanti giochi a più persone adattati ai computer, ma una tale definizione non rende merito alla scelta fatta: è vero che c'è un'inflazione di copie più o meno ben fatte di tutti i giochi che la razza umana ha avuto la pazienza di inventare prima dell'avvento dei computer, ma è anche vero che una parte di tali "giochi" (o forse tutti, chi più chi meno) hanno un valore che supera il semplice aspetto ludico.
I processi intellettivi non avrebbero modo di essere se non fossero convenientemente supportati da una adeguata capacità mnemonica che, per mantenersi in forma, ha però bisogno di un continuo esercizio e anche se ridurre tutto a un gioco può sembrare troppo semplicistico, il negare la possibilità di unire l'utile al dilettevole può ricadere in un errore altrettanto grave: per lo sviluppo della memoria vale più un gioco utile allo scopo che una strofa in rima mal digerita.

Il programma in questione, inoltre, non prevede restrizioni in merito all'età dei giocatori come altri giochi altrettanto validi quali mezzi di supporto alle attività intellettive (carte, scacchi) e si offre, in questo senso, alle più ampie sorprese. Provare per credere. ■

PERSONAL SOFTWARE

Grafici di funzioni trigonometriche

Un programma didattico per Sharp MZ700/731

di *Fabio Farina*
e *Daniele Zampariolo*

Questo programma serve a visualizzare su plotter (presente nello Sharp MZ 731) una qualsiasi funzione trigonometrica precedentemente inserita tramite Input.

Il programma si articola nel seguente modo: dopo la definizione delle 9 funzioni presenti nel programma, a cui è possibile aggiungerne altre, in modo da poter ottenere i più svariati grafici, e dopo aver inserito l'equazione della funzione, il programma inizia col tracciamento in nero delle linee degli assi cartesiani X e Y.

A questo segue il tracciamento più marcato dell'asse X con i relativi valori stampati in verde, che variano da 0 a 360 gradi, con intervalli di 30 gradi ciascuno.

Terminata questa operazione, il plotter traccia nello stesso modo l'asse Y, con i relativi valori ad intervalli di una unità, varianti tra –7 e +7.

Logicamente tali valori non sono fissi, ma possono essere facilmente variati, modificando le routine di verifica di tracciamento dell'asse Y e dei relativi valori e la routine di verifica dei valori Y negativi e positivi.

A questo punto il programma passa a calcolare l'andamento della funzione e in breve tempo inizia il tracciamento in rosso della funzione, inserita partendo da 0 a 360 gradi.

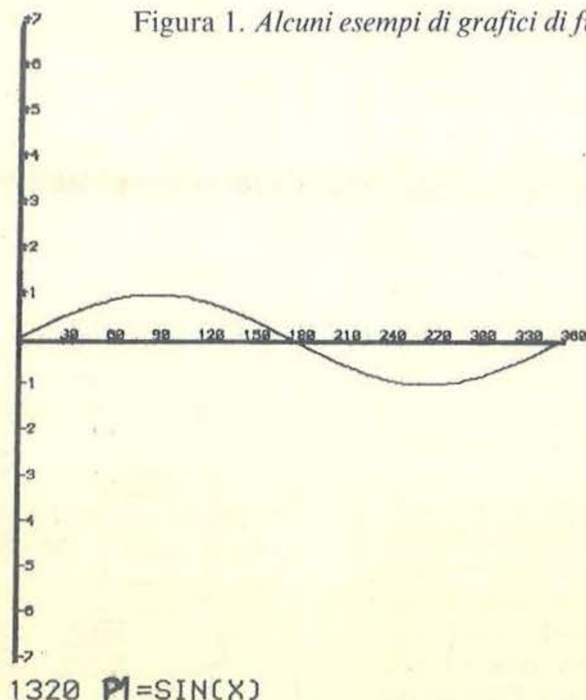
Nel programma è possibile apportare modifiche all'andamento della funzione: ad esempio si può variare l'ampiezza e l'altezza di tale funzione (vedi linee 1340 e 1400).

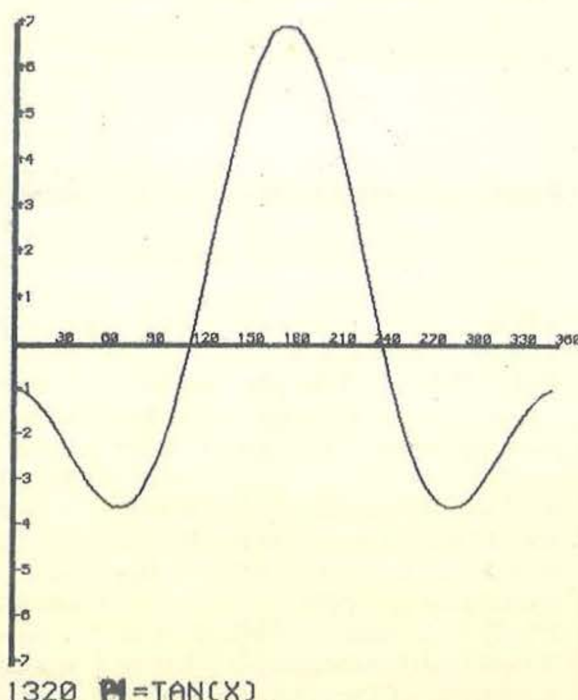
Terminato il grafico, il plotter effettuerà un trascinamento della carta verso l'alto e scriverà nella parte inferiore del foglio l'equazione della funzione disegnata, quindi la penna ritornerà nella posizione iniziale in attesa di un nuovo grafico.

Questo semplice programma, adattabile ad altri computer, può essere notevolmente migliorato, non solo con l'aggiunta di nuove funzioni ma, apportando delle semplici modifiche, ci si può sbizzarrire (si intende per chi ne ha voglia) a calcolare e quindi a trasferire sullo stesso grafico anche le derivate prima e seconda.

Così si possono verificare le capacità, non solo di calcolo, ma anche grafiche di un computer.

Figura 1. *Alcuni esempi di grafici di funzione.*



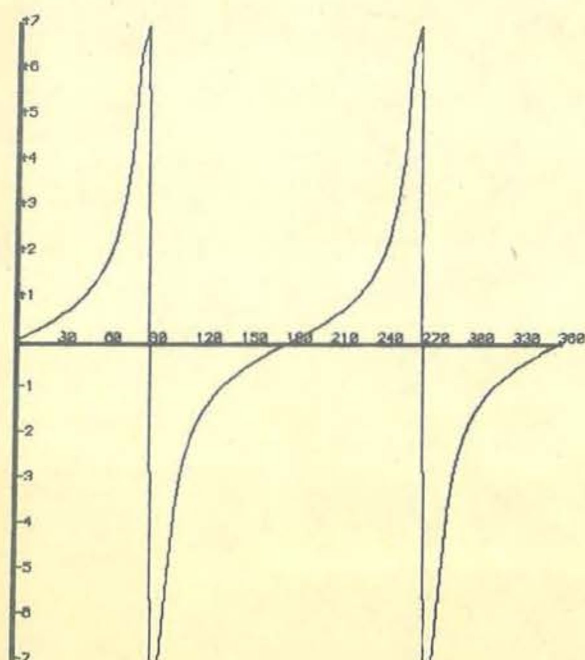


PERSONAL SOFTWARE

Grafici di funzioni trigonometriche

Listato 1. *Il programma per il tracciamento delle funzioni trigonometriche*

```
10 PRINT"▨"
20 CURSOR5,5:PRINT[7,2]"           G R A F
I C O          "
30 CURSOR5,6:PRINT[7,2]"
               "
40 CURSOR5,7:PRINT[7,2]"               D I
               "
50 CURSOR5,8:PRINT[7,2]"
               "
60 CURSOR5,9:PRINT[7,2]"         F U N Z I
 O N E         "
70 CURSOR5,10:PRINT[7,2]"
               "
80 CURSOR5,11:PRINT[7,2]" T R I G O N O
M E T R I C A  "
90 CURSOR16,13:PRINT[7,3]"PRODUCED"
100 CURSOR19,14:PRINT[7,3]"BY"
110 CURSOR14,15:PRINT[7,3]"Z.F SOFTWARE"
120 CURSOR5,17:PRINT[0,7]"          WRI
TTEN           "
130 CURSOR5,18:PRINT[0,7]"
BY             "
140 CURSOR5,19:PRINT[0,7]"           F.F
ARINA          "
150 FORD=4TO6
160 COLOR,,6
170 FOR X=4 TO8
180 FOR Y=4 TO45
190 SET X,Y,D
200 NEXT Y
210 NEXT X
220 FOR X=4 TO8
230 FOR Y=4 TO45
240 SET (X+67),Y,D
250 NEXT Y
260 NEXT X
270 FOR X=4 TO75
280 FOR Y=4 TO8
290 SET X,Y,D
300 NEXT Y
310 NEXT X
320 FOR X=4 TO 75
330 FOR Y=41TO 45
340 SET (X),Y,D
350 NEXT Y
360 NEXT X
370 FOR X=4TO8
380 FOR Y=4 TO45
390 RESET X,Y,D
400 NEXT Y
410 NEXT X
420 FOR X=4 TO8
430 FOR Y=4 TO 45
440 RESET(X+67),Y,D
450 NEXT Y
460 NEXT X
470 FOR X=4 TO 75
480 FOR Y=4 TO 8
490 RESET X,Y,D
500 NEXT Y
510 NEXT X
520 FOR X=4 TO75
530 FOR Y=41 TO45
540 RESET X,Y
550 NEXT Y
560 NEXT X
570 NEXTD
```



Seguito listato Grafici di funzione.

```
580 CLR:COLOR,,7,1:PRINT"Ø"
590 CURSOR3,12:PRINT[0,7]"LA FUNZIONE IM
POSTATA E' NELLA RIGA ":CURSOR4,14:LIST1
320
600 CURSOR8,17:PRINT"LA VUOI CAMBIARE (S
/N)"
610 GETD$
620 IF D$="S"THENGOTO1680
630 IF D$="N"THEN650
640 GOTO610
650 MODE TN
660 COLOR,,7,1
670 PRINT"Ø"
680 SKIP 12
690 REM DEFINIZIONE DELLE FUNZIONI
700 DEFFNSEC(X)=1/COS(X)
710 DEFFNCSC(X)=1/SIN(X)
720 DEFFNCOT(X)=1/TAN(X)
730 DEFFNSH(X)=(EXP(X)-EXP(-X))/2
740 DEFFNCH(X)=(EXP(X)+EXP(-X))/2
750 DEFFNTH(X)=-EXP(-X)/2*(EXP(X)+EXP(-X
))+1
760 DEFFNSCH(X)=2/(EXP(X)+EXP(-X))
770 DEFFNCCH(X)=2/(EXP(X)-EXP(-X))
780 DEFFNCTH(X)=EXP(-X)/2*(EXP(X)-EXP(-X
))+1
790 MODETN
800 PCOLOR0
810 MODE GR:PHOME
820 REM TRACCIAMENTO LINEE ASSI
830 MOVE0,24
840 MOVE0,80:LINE360,80
850 MOVE0,-130:LINE0,290
860 REM TRACCIAMENTO VALORI SULL'ASSE X
870 FORQ=1TO361
880 IFQ=30THENPCOLOR2:GPRINT[0,0],"30"
890 IFQ=60THENPCOLOR2:GPRINT[0,0],"60"
900 IFQ=90THENPCOLOR2:GPRINT[0,0],"90"
910 IFQ=120THENPCOLOR2:GPRINT[0,0],"120"
920 IFQ=150THENPCOLOR2:GPRINT[0,0],"150"
930 IFQ=180THENPCOLOR2:GPRINT[0,0],"180"
940 IFQ=210THENPCOLOR2:GPRINT[0,0],"210"
950 IFQ=240THENPCOLOR2:GPRINT[0,0],"240"
960 IFQ=270THENPCOLOR2:GPRINT[0,0],"270"
970 IFQ=300THENPCOLOR2:GPRINT[0,0],"300"
980 IFQ=330THENPCOLOR2:GPRINT[0,0],"330"
990 IFQ=360THENPCOLOR2:GPRINT[0,0],"360"
1000 MOVEQ,79:LINEQ,81
1010 PCOLOR0
1020 NEXTQ
1030 FORV=-131TO291
1040 PCOLOR0
1050 REM TRACCIAMENTO VALORI SULL'ASSE Y
1060 IF V=-130THENGPRINT[0,0],"-7"

1070 IF V=-100THENGPRINT[0,0],"-6"
1080 IF V=-70THENGPRINT[0,0],"-5"
1090 IF V=-40THENGPRINT[0,0],"-4"
1100 IF V=-10THENGPRINT[0,0],"-3"
1110 IF V=20THENGPRINT[0,0],"-2"
```

più il Personal Computer IBM?

La Libreria Programmi Personal Computer IBM, per esempio.

Il tuo Concessionario IBM per il Personal Computer ha una novità per te: la Libreria Programmi Personal Computer IBM.

La Libreria Programmi Personal Computer IBM è uno strumento utilissimo, che ti permette di trovare nel modo più immediato i programmi firmati da IBM. Puoi guardarli e confrontarli per vedere subito, fra quelli di uno stesso settore applicativo, quale sia il programma che risponde meglio alle tue esigenze di lavoro.

Ma la Libreria Programmi Personal Computer IBM ti dà anche un aiuto in più: con ogni programma troverai la brochure che lo riguarda. Ce ne sono anche molte altre, che ti illustreranno tutto quello che il tuo Personal Computer IBM può fare e le sue applicazioni particolari, come il Videotel.

Puoi prenderle e portarle a casa, per leggerle in tutta tranquillità e pensare bene alla scelta che devi fare.

E non dimenticare che il Concessionario IBM (gli indirizzi sono sulle Pagine Gialle) è un vero esperto, che conosce perfettamente il Personal Computer IBM ed i problemi della tua attività. Potrà consigliarti nel modo migliore nella scelta dei programmi e ti illustrerà tutte le possibilità che il Personal Computer IBM ti offre, oltre alla grande versatilità delle sue prestazioni.

Per acquisto, consulenza e servizi, puoi anche rivolgerti al Negozio IBM Centromilano.

Allora, non ti pare che il Personal Computer IBM abbia veramente qualcosa in più?



Desidero ricevere:
- ☐ Informazioni sul Personal Computer IBM e i suoi programmi.
- ☐ Gli indirizzi dei Concessionari IBM Personal Computer della mia regione
- ☐ Una visita o dimostrazione pratica di un concessionario

Nome e Cognome ....

Azienda ....

La mia attività è ....

Indirizzo ....

Spedisci questo tagliando a: IBM Italia
Direzione Entry Systems
Casella Post. 137 - 20090 Segrate Milano

IBM

PERSONAL SOFTWARE

Grafici di funzioni trigonometriche

Seguito listato Grafici di funzione.

```
1120 IF V=50THENGPRINT[0,0],"-1"
1130 IF V=110THENGPRINT[0,0],"+1"
1140 IF V=140THENGPRINT[0,0],"+2"
1150 IF V=170THENGPRINT[0,0],"+3"
1160 IF V=200THENGPRINT[0,0],"+4"
1170 IF V=230THENGPRINT[0,0],"+5"
1180 IF V=260THENGPRINT[0,0],"+6"
1190 IF V=290THENGPRINT[0,0],"+7"
1200 MOVE0,V:LINE2,V
1210 NEXTV
1220 Z=0:B=0
1230 C1=1:C2=360
1240 FOR V=C1TOC2STEP5
1250 IFABS(C1)>=ABS(C2)THENCR=ABS(C1)
1260 IFABS(C1)<=ABS(C2)THENCR=ABS(C2)
1270 IFB=0THENM=90*4/CR:B=1
1280 X=V:Y=0
1290 X=X*3.1416/180
1300 IFX=0THENX=.000001
1310 REM EQUAZIONE DELLA FUNZIONE
1320 P1=3*COS(2*X)-4*COS(X)
1330 Y=Y+P1
1340 REM*********************************
***********
1350 REM SCHIACCIAMENTO DELLA FUNZIONE
SULL'ASSE X
1360 REM*********************************
***********
1370 Y=30*Y
1380 X=V*M
1390 REM*********************************
******
1400 REM VARIAZIONE DELLA FUNZIONE SULL'
ASSE Y
1410 REM*********************************
******
1420 Y=Y+82
1430 GOSUB1600
1440 IFZ=0THENLINEX,Y:Z=1
1450 LINE X,Y
1460 PCOLOR3
1470 NEXTV
1480 PRINT:LIST1320
1490 GETZ$
1500 PRINT"@"
1510 PHOME
1520 MODE TN
1530 SKIP9
1540 LIST/P1320
1550 SKIP2
1560 PRINT"@@@@@@@@@@@@@@@@@@@@VUOI TERM
INARE (S/N) ?"
1570 GETR1$:IFR1$="S"THENEND
1580 IFR1$<>"N"THEN1570
1590 GOTO580
1600 REM ***********************
1610 REM VERIFICA VALORI DI X E Y
1620 REM ***********************
1630 IFX<0THENX=0
1640 IFX>360THENX=360
1650 IFY<-130THENY=-130

1660 IFY>290THENY=290
1670 RETURN
1680 REM ********************************
**********************
1690 REM INSERIMENTO DELLA FUNZIONE TRIG
ONOMETRICA P1=........
1700 REM ********************************
**********************
1710 CLS:PRINT" INSERISCI LA FUNZIONE DI
 P1 = ":PRINT"@":INPUTF1$
1720 FORI1=1TO10:I2$=CHR$(13)+I2$:NEXT
1730 DEF KEY(5)=I2$
1740 PRINT"@@@@@@@@@@@@@ CONFERMI ? [SI=
'F5' , NO=N]"
1750 GETR1$:IFR1$=CHR$(13)THEN1780
1760 IFR1$<>"N"THEN1750
1770 GOTO1680
1780 COLOR,,0,0:CLS:PRINT"@@@@@1320 ";F1
$
1790 PRINT"GOTO580 "
1800 PRINT"@":DEF KEY(5)="COLOR":STOP
```

REMarks

10-570 - Presentazione programma.
580-640 - Routine - visualizzazione della funzione inserita.
650-780 - Pulizia schermo, trascinamento carta e definizione delle funzioni da parte dell'operatore.
780-1210 - Routine modo grafico con cambio penne, tracciamento assi cartesiani (Plotter) e relativi valori.
1220-1430 - Routine calcolo dell'andamento della curva e variazione sugli assi.
1440-1470 - Routine tracciamento della funzione sulla carta (Plotter).
1480-1500 - Routine di stampa dell'equazione della funzione su carta.
1630-1670 - Routine verifica dei valori X e Y.
1680-1800 - Routine inserimento dell'equazione della funzione.

■

PERSONAL SOFTWARE

Bianchi & Neri per Apple

Un semplice gioco con un computer "intelligente"

di *Stefano Guarinelli*



Le dovute premesse

Il gioco che vi presentiamo è una particolare versione numerica di un "classico" dei giochi di intelligenza: il famosissimo *Master Mind.*

Dopo questa dichiarazione, una parte dei lettori sicuramente esclamerà con una nota di disappunto: "Un'altra versione computerizzata del Master Mind! Tutte le riviste, **Personal Software** inclusa, ne hanno pubblicata almeno una!".

Questo è vero e non lo neghiamo; tuttavia riteniamo che "B&N" meriti ugualmente la vostra attenzione, includendo anche coloro che già annoverano una versione del Master Mind tra i loro giochi per computer. Infatti non è detto che un vecchio problema, anche se affrontato e risolto da tempo, non possa offrire ancora spunti per soluzioni nuove, originali e maggiormente efficaci: ed è in questa direzione che pensiamo di esserci incamminati, elaborando il nostro programma.

Speriamo vivamente che, giunti al termine dell'articolo, anche voi veniate ad essere dello stesso parere.

In caso contrario, bollateci pure come malati cronici di immodestia.

Prima di entrare nel vivo dell'argomento, vorremmo sottolineare che B&N non è riservato esclusivamente ai possessori di Apple II. Con semplici modifiche può infatti girare su un qualsiasi computer dotato di un microprocessore appartenente alla famiglia del 6502: vale a dire Commodore 64, VIC 20, e chi più ne ha, più ne metta.

Facciamo quest'ultima affermazione perché il nostro programma è stato scritto in un BASIC molto generico, privo cioè di istruzioni peculiari della sintassi dell'Apple, e per questo motivo di facile convertibilità. Inoltre, allo scopo di ottenere la massima velocità di esecuzione, la subroutine di gioco del computer è stata realizzata in linguaggio macchina, e quindi l'eventuale lavoro di conversione consisterà solo nello stabilire in quale area di memoria porre le istruzioni e le strutture di dati. Infine, considerando le sue prestazioni, B&N è anche un programma eccezionalmente breve (la parte in BASIC occupa 2.5 Kbyte di memoria, e la subroutine in linguaggio macchina circa 300 byte), quindi di conseguenza alla portata di computer, come il VIC 20 inespanso, che non posseggono grandi capacità di memoria.

Le regole del gioco

B&N è molto vicino ad una tenzone medioevale: vi sono due contendenti e lo scontro ad ogni colpo è sempre più estenuante.

Bianchi e Neri per Apple

Tuttavia si duella a colpi di numeri e non con le armi; la qualità di base non è la forza fisica, bensì l'intelligenza, e lo sfidante del più o meno malcapitato lettore è proprio il suo amico personal, che dimostrerà di essere un avversario tanto disinvolto quanto temibile.
Il gioco consiste in questo: ciascuno dei due contendenti deve formulare segretamente un proprio numero e deve cercare di indovinare per primo quello dell'avversario, onde conseguire la vittoria.

Esistono delle regole precise per la scelta dei numeri: devono essere sempre di quattro cifre, non devono contenere lo zero e ciascuna cifra che li compone non deve essere ripetuta più di una volta. Ad esempio 9.817, 1.436, 2.539 sono corretti, mentre 874, 9.013, 7.527 non lo sono.
Come si fa ad indovinare il numero dell'avversario?
Escludendo l'ipotesi che i due sfidanti siano dotati di telepatia, non resta che la seguente tecnica: a turno i due giocatori formulano, sempre secondo le stesse regole, un altro numero, che ritengono corrispondente a quello dell'avversario, e quest'ultimo è tenuto a dichiarare con correttezza - pena l'ignominia e l'annullamento della partita - che tale numero, rispetto al proprio, vale da uno a quattro Bianchi se sono state indovinate da una a quattro cifre, ma non la loro posizione, e da uno a quattro Neri se non state indovinate da una a quattro cifre ed in più la loro collocazione.

E' spiegata quindi l'origine del nome di questo gioco.

Un altro esempio servirà da ulteriore chiarimento: 1.735, rispetto ad 8.725, dà un punteggio di due Neri e di zero Bianchi, perché il 7 ed il 5 non solo sono presenti in entrambi i numeri, ma occupano anche la stessa posizione, rispettivamente la seconda e la quarta, contando le cifre a partire dalla prima a sinistra. Sempre rispetto ad 8.725, 4.215 vale un Bianco ed un Nero: infatti in 4.215 il 2 è presente, ma non in terza posizione, e vale quindi un Bianco, mentre il 5 continua a valere un Nero, perché ancora in quarta posizione.
Come vedete, le regole del gioco non sono difficili; arduo sarà invece riuscire a vincere il computer.
A conclusione di questa esposizione scherzosa, vorremmo porre una considerazione un po' più seria, che è la seguente.

La regola, secondo la quale in B&N una stessa cifra non può essere ripetuta più di una volta nell'ambito di un numero, potrebbe essere vista come una limitazione che semplifica di molto il gioco, rendendolo più banale rispetto al Master Mind.
Infatti in quest'ultimo, in cui al posto delle cifre numeriche si utilizzano i colori, è invece possibile ripetere fino a quattro volte uno stesso colore.
In realtà, si tratta di una limitazione apparente, perché nel Master Mind si hanno a disposizione solo sette colori diversi, che consentono 2.401 differenti combinazioni di quattro elementi. Invece in B&N si può giocare scegliendo fra 3.024 diversi numeri, consentiti, nonostante la suddetta limitazione, dalle ben nove cifre numeriche disponibili per crearli.
Inoltre, non ripetere più di una volta la stessa cifra, aumenta la quantità di informazioni che si ottengono quando viene comunicato in termini di Bianchi e Neri il risultato di una giocata, lasciando nell'esito di una partita uno spazio minore alla fortuna e maggiore alla strategia e alla capacità di analisi.

In una parola il gioco, oltre che più originale, è anche più avvincente.

Bianchi e Neri con il vostro Apple

Dopo aver esposto le regole del gioco, possiamo descrivere come si comporta il vostro Apple quando esegue B&N, descrizione che vi sarà molto utile nella fase successiva di analisi e comprensione dei listati.
Al fatidico Run, il computer chiederà il vostro nome ed il vostro numero segreto.
Su quest'ultima richiesta i più sospettosi avranno sicuramente da ridire: "come può non vincere un computer a cui è già stato fornito il numero da indovinare?".
In realtà, l'amico personal non tiene per niente conto di questa informazione, ma, come vedremo, accetta il vostro numero segreto solo per stamparlo nel tabellone di gioco.

Alla videata successiva, infatti, nella parte superiore dello schermo compariranno due colonne: una a sinistra, in cui sono riportati automaticamente ad ogni mossa i numeri giocati dal computer, insieme con i risultati realizzati in termini di Bianchi e Neri, ed una a destra, nella quale vengono eseguite le medesime operazioni con i numeri da voi giocati.
E' proprio nella colonna del computer che viene stampato il vostro numero segreto, e a questo punto voi stessi avrete compreso l'utilità di questa operazione.
Avere questo numero incolonnato sopra a quelli giocati dall'amico personal, vi eviterà infatti di ricordarlo a memoria o di doverlo scrivere su un foglio di carta, e vi consentirà di determinare rapidamente il punteggio realizzato dal computer ad ogni sua mossa.

Incominciando una partita, al vostro Apple spetta sempre la prima mossa, e, in accordo con quanto esposto, il primo numero da esso giocato sarà stampato nella colonna con l'intestazione del computer. Tale numero verrà anche riportato nella parte inferiore dello schermo, in cui viene creata una finestra testo riservata a tutti i messaggi di gioco.
Sempre nella finestra testo, il computer chiederà prima quanti Neri e poi quanti Bianchi ha realizzato con quel numero, e spetta a voi fornirgli le risposte esatte.
Come ogni essere dotato di razioci-

nio, il computer rifiuterà risposte scorrette tipo: "ventidue Neri e dieci Bianchi", oppure le lettere o i simboli che potreste battere per distrazione o divertimento.

Nel corso del gioco, potrà accadere di fornire al computer, sempre per distrazione o per diletto, delle risposte sbagliate.

Per esempio potreste rispodere: "zero Neri e due Bianchi" al posto di: "un Nero e un Bianco".

In questo caso il vostro Apple accetterà la risposta, perché compatibile con le regole di B&N.

Quindi fate attenzione a non distrarvi oppure a non divertirvi in questo modo perché l'amico personal, gioca in base ad un algoritmo che, come vedremo, gli consente di valutare la congruenza dell'ultima risposta sulla base di quelle precedenti.

Cosicché, se sbagliate o barate, alla mossa successiva o comunque dopo pochissime mosse, il vostro Apple se ne renderà inesorabilmente conto ed annullerà la partita, comunicandovi un appropriato messaggio.

Sarete costretti dunque a ribattere Run e ad incominciare una nuova partita, cercando di fare maggiore attenzione.

E' questo un fiore all'occhiello del nostro programma. Dopo che il computer avrà fatto la sua prima mossa, verrà il vostro turno.

Sempre nella finestra testo, vi sarà chiesto qual'è il numero da voi giocato.

Anche in questo caso, l'amico personal accetterà solo numeri validi secondo le regole di B&N, e stamperà nella vostra colonna il numero da voi formulato, con a fianco il risultato realizzato.

Se nessuno dei due contendenti ha indovinato il numero segreto dell'altro, - è un po' difficile alla prima mossa - il gioco prosegue ed il vostro Apple prima, e voi poi, giocherete la seconda mossa, sempre secondo le stesse regole.

Il gioco continua fino a quando uno dei due sfidanti indovina il numero segreto dell'avversario.

Se si tratta del computer, aspetterà, prima di proclamare la propria vittoria, che anche voi eseguiate l'ultima mossa.

Il computer, infatti, comincia a giocare sempre per primo e decreterà il pareggio, nel caso anche voi riusciate ad indovinare il suo numero segreto all'ultima mossa.

Se invece indovinate voi per primi, ammetterà immediatamente la sua sconfitta.

In caso di propria vittoria, il computer vi rivelerà sempre il suo numero segreto, affinché anche i più diffidenti fra voi possano controllare i risultati che vi ha fornito e convincersi che non ha barato.

Come potete osservare, il vostro Apple emula in B&N il comportamento di un perfetto giocatore umano e forse anche qualcosa in più.

Infatti, poiché l'algoritmo di gioco è stato scritto in linguaggio macchina e non in BASIC, il vostro amato computer effettua quasi istantaneamente ogni sua mossa, in modo che la durata di una partita sia condizionata solo dalla vostra velocità di gioco.

Infine ricordiamo che il vostro Apple, sempre grazie alla sua procedura di gioco, chiude una partita di B&N al massimo entro sette mosse.

Non sottovalutatelo, dunque, e, quando giocate, impegnatevi o andrete incontro a grosse delusioni.

Analisi del listato BASIC

La parte in BASIC di B&N genera casualmente il numero segreto del computer, i primi due numeri che saranno da esso giocati, cura tutti gli Input e gli Output, nonché le operazioni di verifica su di essi; la stessa parte in BASIC alla subroutine di gioco in linguaggio macchina.

I numeri giocati dall'Apple alla prima ed alla seconda mossa, sono dunque generati dalla subroutine in linguaggio macchina, che pure contiene la procedura di gioco del computer.

Ciò avviene perché questi due numeri hanno solo un valore informativo, e a tale scopo, devono essere formulati casualmente ed in modo che le otto cifre che li costituiscono, siano tutte diverse fra loro.

Ad esempio, se il primo numero giocato è 8.139, il secondo potrà contenere in un ordine qualsiasi quattro delle rimanenti cinque cifre, che sono 2, 4, 5, 6 e 7.

Per formulare rapidamente tali numeri, è sufficiente l'istruzione Randomize del BASIC.

Sulla base dei punteggi realizzati in termini di Bianchi e Neri da questi due numeri, la subroutine in linguaggio macchina opera poi la scelta del successivo numero da giocare.

Può accadere, anche se è un'evenienza abbastanza rara, che il primo numero giocato dal computer realizzi complessivamente, fra Bianchi e Neri, quattro punti: ad esempio 2 Neri e 2 Bianchi o 1 Nero e 3 Bianchi oppure 0 Neri e 4 Bianchi.

In tal caso, il computer ha già identificato quali sono le quattro cifre del vostro numero segreto, e quindi passa alla subroutine in linguaggio macchina direttamente alla seconda mossa.

Infatti, giocare il secondo numero significherebbe in questo caso sprecare una mossa per ottenere un risultato scontato: zero Neri e zero Bianchi.

Osservando il listato BASIC, noterete che non abbiamo abbondato con le istruzioni di carattere grafico, perché abbiamo preferito dare al gioco solo una veste funzionale.

Noterete inoltre l'assenza di istruzioni di carattere sonoro, perché, a nostro avviso, possono infastidire e deconcentrare il giocatore.

Per accontentare, però, quanti fra voi sono patiti del beep, suggeriamo di battere, durante la trascrizione del programma BASIC, Ctrl-G in ogni stringa che riporta un messaggio di gioco.

Tale carattere non sarà visualizzato sullo schermo ed ogni messaggio sarà accompagnato dallo squillare del

Bianchi e Neri per Apple

campanello dell'Apple.
Esaurite le considerazioni di carattere generale, vi forniamo l'analisi del listato BASIC linea per linea.

L'analisi in questione è molto particolareggiata per favorire i non "applisti" nel loro lavoro di conversione.

Sempre ai non "applisti" ricordiamo che le istruzioni seguite da "Rem★★★", sono quelle legate alla subroutine in linguaggio macchina o alle locazioni di memoria a cui essa fa rimerimento. Pertanto queste istruzioni dovranno essere modificate se, nel convertire il programma, la subroutine e/o le locazioni contenenti i dati vengono poste in una differente area di memoria.

10 - Pone come limite alto del programma BASIC l'indirizzo di memoria 24576 ($6000).

150 - Inizializza la variabile V1, utilizzata nella tabulazione verticale del tabellone di gioco.

170-230 - Genera casualmente il numero segreto del computer, memorizzando le sue quattro cifre, tutte diverse fra loro, nel vettore N( ).

240-250 - Carica in memoria la subroutine di gioco in linguaggio macchina, nel caso sia stato dato per la prima volta il Run.

260 - Pulisce lo schermo ed accetta il nome del giocatore come variabile di stringa NM$.

270 - Pulisce lo schermo ed accetta il numero segreto del giocatore come variabile di stringa NS$.

290-300 - Stampa l'intestazione delle due colonne del tabellone di gioco. Nella colonna con l'intestazione del

Lisato 1. *Il programma in BASIC.*

```
 10  HIMEM: 24576: REM  ***
 20  REM      *********************

 30  REM
 40  REM           "BIANCHI & NERI"
 50  REM
 60  REM       BY FRANCO & ANTONIO
 70  REM              MASSARO
 80  REM
 90  REM            VIA MARCONI 42
 100  REM         81026 RECALE (CE)
 110  REM
 120  REM          TEL. 0823/465637
 130  REM
 140  REM     *********************

 150 V1 = 3
 160  REM      IL COMPUTER FORMULA
            IL SUO NUMERO SEGRETO
 170  FOR I = 1 TO 4
 180 N(I) =  INT (9 *  RND (1) + 1
     )
 190  IF I = 1 THEN 230
 200  FOR M = 1 TO I - 1
 210  IF N(M) = N(I) THEN 180
 220  NEXT M
 230  NEXT I
 240  IF  PEEK (900) = 1 THEN 260
 250  PRINT  CHR$ (4);"BLOAD SUBRO
     UTINE": POKE 900,1
 260  HOME : INPUT "COME TI CHIAMI
     ? ";NM$
 270  HOME : INPUT "QUAL' E' IL TU
     O NUMERO? ";NS$
 280  HOME
 290  PRINT "  MOSSE DEL";: HTAB 2
     4: PRINT "MOSSE DI": HTAB 3:
      PRINT "COMPUTER";: HTAB 24:
      PRINT NM$
 300  HTAB 1: PRINT NS$;" NUM.SEGR
     ETO";: HTAB 21: PRINT "....
     NUM.SEGRETO"
 310  REM        GIOCA IL COMPUTER

 320  FOR L = 1 TO 2
 330  FOR I = 1 TO 4
 340 A(L,I) =  INT (9 *  RND (1)) +
     1
 350  IF L = 1 GOTO 390
 360  FOR M = 1 TO 4
 370  IF A(L,I) = A(L - 1,M) GOTO
     340
 380  NEXT M
 390  IF I = 1 GOTO 430
 400  FOR M = 1 TO I - 1
 410  IF A(L,I) = A(L,M) GOTO 340
 420  NEXT M
 430  NEXT I
 440  NEXT L
 450 NUME = 1
 460  POKE 768,1: REM  ***
 470  GOSUB 620
 480  IF NER(NUME) + BIA(NUME) = 4
      GOTO 520
 490 NUME = 2
 500  POKE 768,2: REM  ***
 510  GOSUB 620
 520  REM
 530  CALL 24577: REM  ***
 540 NUME = NUME + 1
 550  POKE 768,NUME: REM  ***
 560  FOR I = 1 TO 4
 570 A(NUME,I) =  PEEK (768 + I): REM
     ***
 580  NEXT I
 590  IF A(NUME,1) = 10 THEN  POKE
     34,0: POKE 35,23: VTAB 18: PRINT
     "HAI SICURAMENTE SBAGLIATO":
      PRINT "A DARMI QUALCHE RISU
     LTATO": PRINT "CONTROLLA LA
     LISTA E ...": PRINT "CHE NON
      SI VERIFICHI PIU'!!!": END

 600  GOSUB 620
 610  GOTO 530
```

Seguito listato BASIC.

```
620  POKE 34,0: POKE 35,15: VTAB
     V1 + 1: HTAB 1: FOR I = 1 TO
     4: PRINT A(NUME,I);: NEXT
630  POKE 34,16: POKE 35,23
640  HOME
650  PRINT "QUESTO E' IL MIO NUME
     RO: "
660  PRINT A(NUME,1);A(NUME,2);A(
     NUME,3);A(NUME,4)
670  PRINT "DAMMI IL RISULTATO"::

680  INPUT "QUANTI NERI? ";NER(NU
     ME)
690  IF NER(NUME) = 4 THEN VI$ =
     "VIN":BIA(NUME) = 0: GOTO 72
     0
700  INPUT "QUANTI BIANCHI? ";BIA
     (NUME)
710 SO = NER(NUME) + BIA(NUME): IF
     SO < 0 OR SO > 4 THEN  HOME
     : PRINT "RIPETI": GOTO 680
720  FOR I = 1 TO 4
730  POKE 773 + (NUME - 1) * 4 +
     I - 1,A(NUME,I): REM  ***
740  NEXT I
750  POKE 809 + (NUME - 1) * 2,NE
     R(NUME): REM  ***
760  POKE 810 + (NUME - 1) * 2,BI
     A(NUME): REM  ***
770  HOME :V1 = V1 + 1: POKE 34,0
     : POKE 35,15: VTAB V1: HTAB
     1: FOR I = 1 TO 4: PRINT A(N
     UME,I);: NEXT : PRINT " ";NE
     R(NUME);"N";BIA(NUME);"B"
780  REM    GIOCA L'AVVERSARIO

790  POKE 34,16: POKE 35,23: HOME
     : INPUT "OK. TOCCA A TE ";AA
     $: IF  LEN (AA$) <  > 4 THEN
     870
800  FOR K = 1 TO 4:SA(K) =  ASC
     ( MID$ (AA$,K,1)): IF SA(K) <
     49 OR SA(K) > 57 THEN 870
810  IF K = 1 THEN 840
820  FOR J = 1 TO K - 1: IF SA(J)
     = SA(K) THEN 870
830  NEXT J
840  NEXT K
850  FOR K = 1 TO 4:T(K) =  VAL (
     MID$ (AA$,K,1)): NEXT
860  GOTO 890
870  HOME : INPUT "RIPETERE ";AA$
     : IF  LEN (AA$) <  > 4 THEN
     870
880  GOTO 800
890  HOME
900 IB = 0:EN = 0
910  FOR I = 1 TO 4
920  FOR L = 1 TO 4
930  IF N(I) <  > T(L) THEN 990
940  IF I <  > L THEN 970
950 EN = EN + 1
960  GOTO 1000
970 IB = IB + 1
980  GOTO 1000
990  NEXT L
1000  NEXT I
1010  POKE 34,0: POKE 35,15: VTAB
      V1: HTAB 21: FOR I = 1 TO 4:
      PRINT T(I);: NEXT : PRINT "
      ";EN;"N";IB;"B"
1020  IF (VI$ = "VIN") * (EN < 4)
      THEN  POKE 34,0: POKE 35,23
      : VTAB 3: HTAB 21: FOR I = 1
      TO 4: PRINT N(I);: NEXT : VTAB
      21: HTAB 1: PRINT "HO VINTO!
      !": PRINT "QUESTO ERA IL MIO
      NUMERO:";: FOR I = 1 TO 4: PRINT
      N(I);: NEXT : END
1030  IF (VI$ <  > "VIN") * (EN =
      4) THEN  POKE 34,0: POKE 35,
      23: VTAB 21: PRINT "HAI VINT
      O ": END
1040  IF (VI$ = "VIN") * (EN = 4)
      THEN  POKE 34,0: POKE 35,23
      : VTAB 21: PRINT "ABBIAMO PA
      REGGIATO": END
1050  RETURN
```

computer viene stampato anche il numero segreto del giocatore. (Noterete poi che la variabile NS$ non verrà più presa in considerazione).

320-440 - Genera casualmente i primi due numeri che devono essere giocati dal computer. Tutte le otto cifre che li compongono sono diverse fra loro e vengono caricate nella matrice bidimensionale A(,).

450 - Pone ad 1 il valore della variabile Nume, contatore nel programma BASIC del numero di mosse giocate dal computer.

460 - Carica 1 nella locazione di memoria 768 ($300), con funzione analoga a Nume nella subroutine in linguaggio macchina.

480 - Invia alla suddetta subroutine direttamente alla seconda mossa se, al ritorno dalla prima, il punteggio realizzato dal computer, fra Bianchi e Neri, è uguale a quattro.

490-500 - Predispone il computer alla seconda mossa.

530 - Invia alla subroutine in linguaggio macchina, caricata in memoria a partire dall'indirizzo 24577 ($6001).

540-550 - Predispone il computer a giocare un'altra mossa.

560-580 - Carica nella matrice A(,) le quattro cifre del numero che sarà giocato dal computer alla mossa indicata dal valore di Nume. Queste cifre, generate dalla subroutine in linguaggio macchina, vengono lette dalle locazioni di memoria che vanno dalla 769 ($301) alla 772 ($304).

Bianchi e Neri per Apple

590 - Se, nella ricerca del numero segreto del giocatore, la subroutine in linguaggio macchina non ha trovato nessun numero valido fra tutti quelli formulabili secondo le regole di B&N, viene segnalato l'errore del giocatore e la partita ha termine.

620 - E' il punto d'inizio di una subroutine BASIC che termina con l'ultima linea del listato. La 620 stampa nella colonna del computer il numero da esso giocato nella mossa indicata dal valore di Nume.

630-660 - Crea la finestra testo della parte inferiore dello schermo, la pulisce e vi ristampa il numero giocato dal computer.

670-680 - Chiede al giocatore il risultato realizzato dal computer e registra il numero dei Neri nel vettore Ner( ).

690 - Se il numero dei Neri è quattro (numero indovinato), la variabile di stringa Vi$ è resa uguale a "Vin", il numero dei Bianchi viene posto automaticamente a zero, viene caricato nel vettore Bia( ) e si salta alla linea 720.

700 - Chiede il numero dei Bianchi totalizzato dal computer e lo carica nel vettore Bia( ).

710 - Verifica la correttezza dei risultati ricevuti, ed in caso negativo ne ripete la procedura di acquisizione.

720-740 - Carica a partire dalla locazione 773 ($305) le cifre dei numeri giocati dal computer. Le locazioni dalla 773 fino alla 808 ($328) inclusa, vengono quindi utilizzate a gruppi di quattro per memorizzare, nell'ordine in cui sono stati formulati, i numeri giocati dal computer.

750-760 - Carica alternativamente, a partire dalla locazione 809 ($329), i Neri e, a partire dalla 810 ($32A), i Bianchi realizzati dal computer giocando i propri numeri. Lo spazio in memoria riservato a tale scopo arriva fino alla locazione 826 ($33A) inclusa.

770 - Pulisce la finestra testo e stampa nella colonna del computer il numero giocato dal computer stesso con a fianco il relativo punteggio.

790 - Pulisce la finestra testo. Segnala al giocatore che deve eseguire la sua mossa ed accetta il numero formulato da quest'ultimo come variabile di stringa AA$. Se la lunghezza di AA$ non è di quattro elementi (Input scorretto), ripete la procedura di richiesta del numero.

800-880 - Carica i codici ASCII dei quattro elementi di AA$ nel vettore SA( ). Se tali codici corrispondono alle cifre numeriche da 1 a 9 e se non sono ripetuti più di una volta, carica il loro valore numerico nel vettore T( ). In caso contrario ripete l'intera procedura di richiesta del numero.

900-1000 - Calcola il punteggio realizzato dal giocatore ed assegna alle variabili EN ed IB rispettivamente i valori corrispondenti al numero dei Neri e dei Bianchi realizzati nel confronto.

1010 - Stampa il numero formulato dal giocatore nella sua colonna, con a fianco il punteggio realizzato.

1020 - Se il computer ha realizzato quattro Neri (Vi$="Vin") e se il giocatore non ha totalizzato questo punteggio, viene annunciata la vittoria del computer, ed il suo numero segreto viene stampato sia nella finestra testo, che nella intestazione della colonna del giocatore. La partita ha termine.

1030 - Se avviene il contrario della linea precedente, viene decretata la vittoria del giocatore e termina la partita.

1040 - Se sia il computer che il giocatore hanno realizzato quattro Neri, viene proclamato il pareggio e la partita ha termine.

Listato 2. *I codici esadecimali.*

```
*6001.60DB

6001- A9 00 8D 01 03 EE 01
6008- 03 AD 01 03 C9 0A 30 01
6010- 60 A9 00 8D 02 03 EE 02
6018- 03 AD 02 03 C9 0A F0 E6
6020- A9 00 8D 03 03 EE 03 03
6028- AD 03 03 C9 0A F0 E7 A9
6030- 00 8D 04 03 EE 04 03 AD
6038- 04 03 C9 0A F0 E7 A2 00
6040- AD 04 03 DD 01 03 F0 EC
6048- E8 E0 03 D0 F6 AD 03 03
6050- CD 01 03 F0 D0 CD 02 03
6058- F0 CB AD 02 03 CD 01 03
6060- F0 B4 A9 04 8D 3B 03 A0
6068- 00 8C 40 03 EE 40 03 8C
6070- 3F 03 8C 3E 03 98 AE 40
6078- 03 CA F0 07 18 6D 3B 03
6080- 4C 79 60 AA 8E 3C 03 BD
6088- 05 03 D9 01 03 D0 03 20
6090- C7 60 C8 C0 04 D0 F0 A0
6098- 00 E8 8A 38 ED 3C 03 C9
60A0- 04 D0 E4 AD 40 03 18 6D
60A8- 40 03 AA BD 27 03 CD 3E
60B0- 03 D0 81 BD 28 03 CD 3F
60B8- 03 D0 09 AD 40 03 CD 00
60C0- 03 D0 A9 60 4C 34 60 8A
60C8- 38 ED 3C 03 8D 3D 03 CC
60D0- 3D 03 F0 04 EE 3F 03 60
60D8- EE 3E 03 60
```

La subroutine in linguaggio macchina

Siamo finalmente giunti all'analisi della subroutine in linguaggio macchina. Per identificare il numero segreto del suo avversario, il computer fa riferimento ad un numero, battezzato: "in corso".

Questo numero assume, come primo valore, 1.234 - il più basso formulabile secondo le regole di B&N - e viene supposto sempre uguale al numero segreto del giocatore umano.

Si tratta naturalmente solo di una supposizione, che bisogna quindi verificare.

A tale scopo, 1.234 viene paragonato ai numeri giocati in precedenza dal computer stesso. I punteggi ottenuti, relativi cioè al numero in corso, sono a loro volta confrontati con quelli corrispondenti di gioco.

Se nel confronto risultano tutti uguali, 1.234 ha elevate probabilità di essere il numero segreto ricercato dal computer, e pertanto viene giocato.

Se invece questa condizione non si verifica, il numero in corso è sicuramente diverso da quello segreto.

Il suo valore viene allora sostituito da quello immediatamente successi-



Bianchi e Neri per Apple

vo secondo le regole di B&N. In altre parole, viene sostituito da 1.235 e, se anche quest'ultimo non dovesse risultare valido, da 1.236 e così via. Il valore del numero in corso viene dunque incrementato fino a quando non risulterà giocabile.
E' intuitivo che, scorrendo i numeri di B&N in senso crescente, il computer dovrà necessariamente individuare prima o poi il numero segreto del suo avversario.

Matematicamente, è dimostrabile che, adottando questa procedura, il vostro personal raggiunge il suo obiettivo al massimo in sette mosse.
Vi proponiamo adesso un quesito. Può accadere che il numero in corso assuma tutti i valori numerici di B&N, senza risultare mai giocabile? La risposta è sì, ma solo nel caso sia stato comunicato al computer un punteggio di gioco errato.
E' ora chiarito anche il modo in cui il vostro Apple valuta la congruenza dei punteggi che gli vengono forniti nel corso di una partita.
Da quanto detto, emerge la facile considerazione che, giocando contro il computer, conviene usare numeri segreti con valori alti: vale a dire numeri che iniziano con 7, 8 o 9. Infatti, se la subroutine ricerca i numeri da giocare in senso crescente, quelli più elevati saranno presi in considerazione per ultimi.

Questa tattica di gioco vi concederà un lieve vantaggio, che tuttavia non rappresenta un handicap per il vostro personal, in grado comunque di chiudere una qualsiasi partita al massimo entro la settima mossa.
Un giocatore umano, invece, per quanto bravo, non può risparmiarsi, in certi casi, di terminare una partita in meno di otto mosse.
Il listato Assembly, che fra l'altro è stato pensato e commentato in maniera modulare, dovrebbe risultare piuttosto comprensibile: esaminandolo, noterete che abbiamo fatto largo uso dell'indirizzamento indicizzato, grazie al quale abbiamo potuto ridurre di molto la lunghezza della subroutine.
A questo punto, possiamo già congedarci dagli "applisti", che, dopo aver caricato in memoria tramite il monitor i codici esadecimali della subroutine, possono già salvarla su disco battendo:

BSAVE SUBROUTINE, A$6001, L219.

Agli alieni, cioè ai non "applisti", sono destinate le nostre ultime parole.
Come abbiamo già premesso, il problema della conversione è legato solo all'area di memoria in cui la subroutine ed i dati a cui essa fa riferimento devono essere collocati.
Se la collocazione, come è molto probabile, è diversa da quella dell'Apple, bisognerà cambiare i campi degli indirizzi. Questa operazione, semplicissima se possedete un assemblatore, diventa un po' più difficoltosa se ne siete sprovvisti.
Non è però impossibile, e, per incoraggiarvi e consolarvi, vi ricordiamo che il campo di spostamento di istruzioni usate a profusione, tipo Bne, Beq e Bmi, non deve essere modificato.

Questa lunga tortura è ora - finalmente! - terminata e auguriamo a tutti buon divertimento. ■

PERSONAL SOFTWARE

CMK 49: note e musica con il Commodore 64

Capita sempre più di frequente trovare sul mercato dell'informatica, apparecchi che uniti ad un certo computer, sfruttano le sue doti musicali. Il C 64 è uno dei personal più diffusi e con le migliori capacità sonore grazie al SID, che è un piccolo sintetizzatore facente parte dell'hardware di questa macchina ed è in grado di fornire numerosi timbri. E' quindi chiaro che la maggior parte di prodotti "informatico-musicale" tenda ad abbinarsi con questo computer.

La SIEL è uno dei costruttori che ha fatto i maggiori sforzi per realizzare buoni package e strumenti musicali orientati verso quelle persone che disponendo di un computer, ma non essendo programmatori tanto esperti da "farlo suonare", vogliono soddisfare il loro desiderio di musica. Fra i tanti prodotti di questa casa c'è una tastiera, il CMK 49, che può essere collegata al Commodore 64 e, attraverso un programma, permette di suonare intervenendo sul sintetizzatore interno di cui abbiamo accennato prima. Vediamo, dunque, alcune delle sue interessanti caratteristiche: connettendo il CMK 49 al computer è possibile avere il controllo di altri strumenti musicali, purché dotati dell'interfaccia MIDI. Sulla scatola che collega la tastiera al C 64 c'è un connettore multiplo il quale permette di rilanciare i controlli provenienti dalla tastiera (unione per l'interfaccia MIDI). E' attraverso la cartuccia da inserire sul retro del computer che il CMK 49 viene a questi collegato. La tastiera è composta da quattro ottave; non è presente in essa alcun interruttore o manopola poiché ogni regolazione viene effettuata tramite il programma, quindi sui tasti del Commodore. Tale programma è disponibile tanto su cassetta, quanto su floppy-disk.



Il menu principale si compone di sette opzioni:

1) Edit Play;
2) Polyphonic New Sound;
3) Monophonic New Sound;
4) Midi Master Keyboard;
5) Tape Operation;
6) End of Job;
7) Demo Song;

Analizziamole partendo dal fondo: la settima presenta all'audience alcuni motivi per dare sfoggio delle capacità musicali di tutto il dispositivo. Per concludere il lavoro si selezionerà la numero sei. Attraverso Tape Operation si ha accesso al dispositivo di memorizzazione dei dati collegato: da qui si potranno caricare i suoni presentati dal fabbricante (ce ne sono 40) o dall'utente stesso, che può registrare i timbri da lui creati.

Con MIDI Master Keyboard è possibile controllare attraverso l'interfaccia MIDI, altre tastiere suonando sul CMK 49. La seconda e terza opzione permettono la creazione di nuovi timbri: c'è tutta una serie di parametri che possono essere controllati e la cui selezione avviene attraverso una freccia rossa. La selezione dei dati sui quali è possibile intervenire e le modifiche successive sono molto chiare e pratiche: si possono cambiare rapidamente i timbri che, come già detto, possono essere memorizzati.

L'Edit Play, infine, presenta l'elenco di tutti i timbri che possono essere reperiti nella memoria partendo da quelli già predisposti dal costruttore.

Quando il programma è caricato scompare dallo schermo il colore blu per riapparire più tardi con una fascia: questo vuol significare per l'utente che la tastiera è abilitata per eseguire il timbro selezionato. Il sintetizzatore del Commodore 64 dispone di tre oscillatori che sono stati tutti utilizzati allo scopo di produrre ogni timbro udibile; è chiaro che nell'ambito monofonico i suoni saranno più ricchi che nel caso polifonico, ma anche qui non ci si potrà proprio lamentare. L'estrema chiarezza delle due opzioni Mono/Polyphonic New Sound permettono di utilizzare il CMK 49 con funzione didattica dando la possibilità all'utente di controllare le variazioni apportate ad un timbro dopo la modifica di ogni parametro.

La diffusione del C 64 e la bontà di questo prodotto fanno del CMK 49 un apparecchio molto interessante: il software in dotazione mette in risalto le capacità sonore del computer aiutando l'utente a scoprire le infinite potenzialità del SID. ■



Grendel, il vostro renumber

Un'utility per Spectrum

di *Stefano Cerutti*

Così Grendel distrusse e devastò tutto il reame, uno contro tutti, in una lotta malefica, finchè la sala reale fu deserta e desolata.
In questo modo vengono descritte le azioni di Grendel, un orrendo mostro che verrà poi ucciso da Beowulf, nel più antico poema della letteratura anglosassone intitolato appunto "Beowulf" e composto probabilmente nel settimo secolo da un bardo anonimo.

Forse si sta già destando in voi il desiderio di rivivere con questo videogioco le gesta dell'impavido cavaliere e far mordere la polvere a Grendel, l'infernale divoratore di guerrieri?
Sono spiacente di dovervi dare una delusione, ma il programma che verrà presentato in queste pagine appartiene alla categoria delle utility, e deve il suo nome bizzarro alle funzioni che assolve, infatti GRENDEL significa "General RENumber e line DELeter".
In pratica, avete sottomano un renumeratore totale e uno strumento efficace per eliminare interi gruppi di linee BASIC evitando di cancellarle una per una.
Terminata l'introduzione, veniamo quindi al sodo: il povero Grendel può starsene tranquillo!

Come caricare Grendel

Per prima cosa occorre digitare il listato 1 oppure 2, rispettivamente per Grendel 48 e Grendel 16, e salvarlo poi su nastro.
La linea zero contenente le due Def Fn... va inserita come linea uno e poi sistemata con Poke 23756, 0; essendosi assicurati di avere copiato tutto correttamente, si può dare Save "Grendel" Line 0.
Se la linea uno non è diventata linea zero, bisogna sostituire il valore della variabile di sistema Prog (byte 23635-6) al numero 23.756.
Dopo aver verificato e resettato, si inizia a digitare attentamente il listato 3 oppure 4 e si dà un Run: occorre poi inserire ad uno ad uno (sigh!) i 653 codici decimali di figura 1 oppure 2.
Se non vengono segnalati errori, verrà prodotto il file da registrare subito dopo Grendel;

Come utilizzare Grendel

Se fosse necessario cancellare le linee, per esempio, comprese tra cinquecento e ottocento, dando Print Fn d (500, 800) si risolverebbe la questione in una frazione di microsecondo.
Occorre sempre tenere presente che i numeri di linea di inizio e di fine sono inclusi nella cancellazione, e devono sempre essere realmente esistenti, altrimenti verrà prodotto un messaggio di errore.

Allo stesso risultato porta l'inserimento di due numeri non consecutivi: ad esempio, Print Fn d (1100,

Listato 1. *Grendel bootstrap loader 48 Kbyte.*

```
   0>DEF FN d(D,R)=USR 64700   :
     DEF FN r(R,D)=USR 64822   :
 REM
         Grendel 48K
     © 1985  by  Robin Drake

  10 CLEAR 64699: LOAD ""CODE
  20 PRINT " Per cancellare grup
pi di linee adiacenti, dare un:"
''"       PRINT FN d(start,end)"
  30 PRINT ''"Per rinumerare il
programma dare"''"  PRINT FN r(s
tart,incremento)"
  40 PRINT ''"Tenere presente ch
e la linea '0' deve sempre esser
e in memoria  per permettere il
funzionamento dei due algoritmi.
"
```

Listato 2. *Grendel bootstrap loader 16 Kbyte.*

```
   0>DEF FN d(D,R)=USR 31940
     DEF FN r(R,D)=USR 32062
 REM
         Grendel 16K
     © 1985  by  Robin Drake

  10 CLEAR 31939: LOAD ""CODE
  20 PRINT " Per cancellare grup
pi di linee adiacenti, dare un:"
''"       PRINT FN d(start,end)"
  30 PRINT ''"Per rinumerare il
programma dare"''"  PRINT FN r(s
tart,incremento)"
  40 PRINT ''"Tenere presente ch
e la linea '0' deve sempre esser
e in memoria  per permettere il
funzionamento dei due algoritmi.
"
```

Grendel, il vostro renumber

Listato 3. *Caricatore decimale per Grendel 48.*

```
   1 REM  {}{}{}{}{}{}{}{}{}{}{}
   2 REM  {                    }
   3 REM  {  Caricatore BASIC  }
   4 REM  {                    }
   5 REM  {    del programma   }
   6 REM  {                    }
   7 REM  {     GRENDEL 48     }
   8 REM  {                    }
   9 REM  {}{}{}{}{}{}{}{}{}{}{}
  10 CLEAR 64699: LET n=2410: LE
T a=64700
  20 LET u=0: FOR f=a TO a+19: I
NPUT (f),s: POKE f,s: LET u=u+s:
 PRINT f,s: NEXT f: BEEP 1,30
  30 IF n<>u THEN PRINT "Errore
nei byte compresi tra ";a;" e ";
f-1: STOP
  40 READ n: IF n=2↑16 THEN BEEP
 10.4,20: PRINT #1;"Fai partire
il registratore, poipremi un tas
to.": POKE 23624,63: SAVE "GrenC
ODE 48K"CODE 64700,653: POKE 236
24,56: PRINT #1;"Riavvolgi il na
stro per verifica": VERIFY "Gren
CODE 48K"CODE 64700: GO TO 32767
  50 LET a=f: GO TO 20
  60 DATA 2160,2280,2650,2776,20
19,2873,2514,2181,2329,1939,2055
,1949,3357,2931,2739,2119,2169,2
970,2988,2741,1610,2323,2541,259
7,2115,1840,2201,1592,3016,1680,
1798,762,65536
  70 REM  © 1985 by Robin Drake
```

Listato 4. *Caricatore decimale per Grendel 16.*

```
   1 REM  {}{}{}{}{}{}{}{}{}{}{}
   2 REM  {                    }
   3 REM  {  Caricatore BASIC  }
   4 REM  {                    }
   5 REM  {    del programma   }
   6 REM  {                    }
   7 REM  {     GRENDEL 16     }
   8 REM  {                    }
   9 REM  {}{}{}{}{}{}{}{}{}{}{}
  10 CLEAR 31939: LET n=2170: LE
T a=31940
  20 LET u=0: FOR f=a TO a+19: I
NPUT (f),s: POKE f,s: LET u=u+s:
 PRINT f,s: NEXT f: BEEP 1,30
  30 IF n<>u THEN PRINT "Errore
nei byte compresi tra ";a;" e ";
f-1: STOP
  40 READ n: IF n=2↑16 THEN BEEP
 10.4,20: PRINT #1;"Fai partire
il registratore, poipremi un tas
to.": POKE 23624,63: SAVE "GrenC
ODE 16K"CODE 31940,653: POKE 236
24,56: PRINT #1;"Riavvolgi il na
stro per verifica": VERIFY "Gren
CODE 16K"CODE 31940: GO TO 32767
  50 LET a=f: GO TO 20
  60 DATA 2160,2280,2650,2656,18
99,2513,2394,2061,2209,1699,1815
,1829,3117,2931,2619,1999,2169,2
730,2876,2493,1490,2203,2421,259
7,2115,1840,2201,1592,2536,1680,
1678,762,65536
  70 REM  © 1985 by Robin Drake
```

Figura 1. *Codici decimali da inserire con il listato 3 (48 Kbyte).*

64700	205	64739	1	64778	225	64817	35
64701	22	64740	45	64779	205	64818	78
64702	253	64741	48	64780	110	64819	35
64703	239	64742	2	64781	25	64820	70
64704	3	64743	207	64782	205	64821	201
64705	55	64744	10	64783	184	64822	205
64706	56	64745	197	64784	25	64823	22
64707	205	64746	225	64785	205	64824	253
64708	162	64747	227	64786	232	64825	205
64709	45	64748	229	64787	25	64826	162
64710	203	64749	197	64788	207	64827	45
64711	9	64750	225	64789	255	64828	237
64712	48	64751	205	64790	42	64829	67
64713	2	64752	110	64791	11	64830	68
64714	207	64753	25	64792	92	64831	255
64715	11	64754	40	64793	205	64832	205
64716	205	64755	2	64794	42	64833	162
64717	22	64756	207	64795	253	64834	45
64718	253	64757	9	64796	229	64835	237
64719	205	64758	35	64797	205	64836	67
64720	1	64759	35	64798	182	64837	70
64721	1	64760	227	64799	42	64838	255
64722	1	64761	94	64800	225	64839	42
64723	1	64762	35	64801	35	64840	83
64724	1	64763	86	64802	35	64841	92
64725	1	64764	35	64803	205	64842	126
64726	1	64765	25	64804	42	64843	230
64727	1	64766	209	64805	253	64844	192
64728	1	64767	213	64806	205	64845	194
64729	1	64768	167	64807	182	64846	42
64730	1	64769	237	64808	42	64847	255
64731	1	64770	82	64809	201	64848	1
64732	1	64771	43	64810	35	64849	255
64733	1	64772	43	64811	35	64850	255
64734	1	64773	235	64812	126	64851	126
64735	1	64774	225	64813	35	64852	230
64736	1	64775	115	64814	94	64853	192
64737	1	64776	35	64815	35	64854	32
64738	1	64777	114	64816	86	64855	10



Seguito figura 1.

64856	35	64947	255
64857	35	64948	237
64858	94	64949	91
64859	35	64950	69
64860	86	64951	92
64861	35	64952	25
64862	3	64953	34
64863	25	64954	69
64864	24	64955	92
64865	241	64956	175
64866	205	64957	50
64867	43	64958	71
64868	45	64959	92
64869	237	64960	231
64870	75	64961	253
64871	68	64962	52
64872	255	64963	13
64873	205	64964	254
64874	43	64965	58
64875	45	64966	40
64876	205	64967	248
64877	233	64968	254
64878	52	64969	13
64879	56	64970	202
64880	27	64971	2
64881	239	64972	255
64882	4	64973	254
64883	56	64974	234
64884	237	64975	202
64885	75	64976	58
64886	70	64977	255
64887	255	64978	254
64888	205	64979	225
64889	43	64980	40
64890	45	64981	27
64891	205	64982	254
64892	233	64983	229
64893	52	64984	40
64894	56	64985	82
64895	12	64986	254
64896	239	64987	236
64897	15	64988	40
64898	56	64989	78
64899	205	64990	254
64900	162	64991	237
64901	45	64992	40
64902	33	64993	74
64903	240	64994	254
64904	216	64995	240
64905	9	64996	40
64906	48	64997	11
64907	12	64998	254
64908	205	64999	247
64909	197	65000	40
64910	22	65001	66
64911	33	65002	254
64912	1	65003	248
64913	0	65004	40
64914	34	65005	35
64915	68	65006	195
64916	255	65007	242
64917	34	65008	254
64918	70	65009	205
64919	255	65010	119
64920	42	65011	0
64921	70	65012	254
64922	255	65013	35
64923	167	65014	32
64924	237	65015	55
64925	91	65016	205
64926	68	65017	182
64927	255	65018	24
64928	237	65019	254
64929	82	65020	13
64930	34	65021	202
64931	69	65022	2
64932	92	65023	255
64933	42	65024	254
64934	83	65025	58
64935	92	65026	40
64936	35	65027	188
64937	35	65028	254
64938	35	65029	59
64939	34	65030	40
64940	176	65031	36
64941	92	65032	254
64942	34	65033	44
64943	93	65034	40
64944	92	65035	32
64945	42	65036	205
64946	68	65037	119

65038	0	65129	34
65039	24	65130	93
65040	231	65131	92
65041	205	65132	213
65042	119	65133	205
65043	0	65134	114
65044	254	65135	254
65045	34	65136	24
65046	40	65137	50
65047	3	65138	42
65048	35	65139	83
65049	24	65140	92
65050	7	65141	237
65051	205	65142	91
65052	119	65143	70
65053	0	65144	255
65054	254	65145	126
65055	34	65146	230
65056	32	65147	192
65057	249	65148	32
65058	205	65149	11
65059	119	65150	126
65060	0	65151	184
65061	254	65152	56
65062	202	65153	16
65063	194	65154	192
65064	242	65155	35
65065	254	65156	126
65066	24	65157	185
65067	0	65158	56
65068	205	65159	11
65069	119	65160	201
65070	0	65161	33
65071	229	65162	240
65072	209	65163	216
65073	254	65164	25
65074	13	65165	208
65075	202	65166	17
65076	2	65167	15
65077	255	65168	39
65078	254	65169	201
65079	58	65170	35
65080	40	65171	229
65081	134	65172	42
65082	254	65173	68
65083	32	65174	255
65084	40	65175	25
65085	238	65176	235
65086	205	65177	225
65087	182	65178	197
65088	24	65179	35
65089	40	65180	78
65090	19	65181	35
65091	254	65182	70
65092	48	65183	35
65093	218	65184	9
65094	242	65185	193
65095	254	65186	24
65096	254	65187	213
65097	58	65188	193
65098	210	65189	213
65099	242	65190	42
65100	254	65191	93
65101	205	65192	92
65102	119	65193	229
65103	0	65194	213
65104	254	65195	197
65105	32	65196	209
65106	32	65197	205
65107	234	65198	229
65108	24	65199	25
65109	247	65200	34
65110	254	65201	93
65111	13	65202	92
65112	40	65203	193
65113	7	65204	205
65114	254	65205	43
65115	58	65206	45
65116	194	65207	42
65117	242	65208	93
65118	254	65209	92
65119	24	65210	34
65120	0	65211	91
65121	43	65212	92
65122	43	65213	42
65123	70	65214	81
65124	43	65215	92
65125	78	65216	229
65126	43	65217	62
65127	43	65218	255
65128	43	65219	205



Grendel, il vostro renumber

Seguito figura 1.

65220	1	65255	42	65290	83	65325	33
65221	22	65256	93	65291	92	65326	0
65222	205	65257	92	65292	58	65327	0
65223	227	65258	35	65293	68	65328	34
65224	45	65259	35	65294	255	65329	69
65225	225	65260	35	65295	237	65330	92
65226	205	65261	113	65296	91	65331	62
65227	21	65262	35	65297	70	65332	1
65228	22	65263	112	65298	255	65333	50
65229	42	65264	35	65299	114	65334	71
65230	91	65265	35	65300	35	65335	92
65231	92	65266	34	65301	115	65336	207
65232	34	65267	93	65302	35	65337	255
65233	93	65268	92	65303	78	65338	42
65234	92	65269	14	65304	35	65339	176
65235	42	65270	0	65305	70	65340	92
65236	176	65271	126	65306	35	65341	70
65237	92	65272	22	65307	9	65342	43
65238	70	65273	1	65308	235	65343	78
65239	43	65274	205	65309	6	65344	9
65240	78	65275	154	65310	0	65345	35
65241	235	65276	25	65311	79	65346	24
65242	42	65277	254	65312	9	65347	190
65243	93	65278	13	65313	235	65348	10
65244	92	65279	194	65314	126	65349	0
65245	9	65280	192	65315	230	65350	10
65246	193	65281	253	65316	192	65351	0
65247	167	65282	35	65317	32	65352	201
65248	237	65283	126	65318	3	65353	0
65249	66	65284	230	65319	121	65354	0
65250	235	65285	192	65320	24	65355	0
65251	115	65286	202	65321	233	65356	0
65252	35	65287	168	65322	34	65357	0
65253	114	65288	253	65323	68	65358	0
65254	193	65289	42	65324	255	65359	0

Figura 2. *Codici decimali da inserire con il listato 4 (16 Kbyte).*

31940	205	31988	229	32036	229	32084	192
31941	30	31989	197	32037	205	32085	194
31942	125	31990	225	32038	182	32086	50
31943	239	31991	205	32039	42	32087	127
31944	3	31992	110	32040	225	32088	1
31945	55	31993	25	32041	35	32089	255
31946	56	31994	40	32042	35	32090	255
31947	205	31995	2	32043	205	32091	126
31948	162	31996	207	32044	50	32092	230
31949	45	31997	9	32045	125	32093	192
31950	203	31998	35	32046	205	32094	32
31951	9	31999	35	32047	182	32095	10
31952	48	32000	227	32048	42	32096	35
31953	2	32001	94	32049	201	32097	35
31954	207	32002	35	32050	35	32098	94
31955	11	32003	86	32051	35	32099	35
31956	205	32004	35	32052	126	32100	86
31957	30	32005	25	32053	35	32101	35
31958	125	32006	209	32054	94	32102	3
31959	205	32007	213	32055	35	32103	25
31960	162	32008	167	32056	86	32104	24
31961	45	32009	237	32057	35	32105	241
31962	48	32010	82	32058	78	32106	205
31963	2	32011	43	32059	35	32107	43
31964	207	32012	43	32060	70	32108	45
31965	10	32013	235	32061	201	32109	237
31966	197	32014	225	32062	205	32110	75
31967	225	32015	115	32063	30	32111	76
31968	205	32016	35	32064	125	32112	127
31969	110	32017	114	32065	205	32113	205
31970	25	32018	225	32066	162	32114	43
31971	40	32019	205	32067	45	32115	45
31972	2	32020	110	32068	237	32116	205
31973	207	32021	25	32069	67	32117	233
31974	9	32022	205	32070	76	32118	52
31975	35	32023	184	32071	127	32119	56
31976	35	32024	25	32072	205	32120	27
31977	229	32025	205	32073	162	32121	239
31978	205	32026	232	32074	45	32122	4
31979	162	32027	25	32075	237	32123	56
31980	45	32028	207	32076	67	32124	237
31981	48	32029	255	32077	78	32125	75
31982	2	32030	42	32078	127	32126	78
31983	207	32031	11	32079	42	32127	127
31984	10	32032	92	32080	83	32128	205
31985	197	32033	205	32081	92	32129	43
31986	225	32034	50	32082	126	32130	45
31987	227	32035	125	32083	230	32131	205

Seguito figura 2.

32132	233	32223	229	32314	13	32405	208
32133	52	32224	40	32315	202	32406	17
32134	56	32225	82	32316	10	32407	15
32135	12	32226	254	32317	127	32408	39
32136	239	32227	236	32318	254	32409	201
32137	15	32228	40	32319	58	32410	35
32138	56	32229	78	32320	40	32411	229
32139	205	32230	254	32321	134	32412	42
32140	162	32231	237	32322	254	32413	76
32141	45	32232	40	32323	32	32414	127
32142	33	32233	74	32324	40	32415	25
32143	240	32234	254	32325	238	32416	235
32144	216	32235	240	32326	205	32417	225
32145	9	32236	40	32327	182	32418	197
32146	48	32237	11	32328	24	32419	35
32147	12	32238	254	32329	40	32420	78
32148	205	32239	247	32330	19	32421	35
32149	197	32240	40	32331	254	32422	70
32150	22	32241	66	32332	48	32423	35
32151	33	32242	254	32333	218	32424	9
32152	1	32243	248	32334	250	32425	193
32153	0	32244	40	32335	126	32426	24
32154	34	32245	35	32336	254	32427	213
32155	76	32246	195	32337	58	32428	193
32156	127	32247	250	32338	210	32429	213
32157	34	32248	126	32339	250	32430	42
32158	78	32249	205	32340	126	32431	93
32159	127	32250	119	32341	205	32432	92
32160	42	32251	0	32342	119	32433	229
32161	78	32252	254	32343	0	32434	213
32162	127	32253	35	32344	254	32435	197
32163	167	32254	32	32345	32	32436	209
32164	237	32255	55	32346	32	32437	205
32165	91	32256	205	32347	234	32438	229
32166	76	32257	182	32348	24	32439	25
32167	127	32258	24	32349	247	32440	34
32168	237	32259	254	32350	254	32441	93
32169	82	32260	13	32351	13	32442	92
32170	34	32261	202	32352	40	32443	193
32171	69	32262	10	32353	7	32444	205
32172	92	32263	127	32354	254	32445	43
32173	42	32264	254	32355	58	32446	45
32174	83	32265	58	32356	194	32447	42
32175	92	32266	40	32357	250	32448	93
32176	35	32267	188	32358	126	32449	92
32177	35	32268	254	32359	24	32450	34
32178	35	32269	59	32360	0	32451	91
32179	34	32270	40	32361	43	32452	92
32180	176	32271	36	32362	43	32453	42
32181	92	32272	254	32363	70	32454	81
32182	34	32273	44	32364	43	32455	92
32183	93	32274	40	32365	78	32456	229
32184	92	32275	32	32366	43	32457	62
32185	42	32276	205	32367	43	32458	255
32186	76	32277	119	32368	43	32459	205
32187	127	32278	0	32369	34	32460	1
32188	237	32279	24	32370	93	32461	22
32189	91	32280	231	32371	92	32462	205
32190	69	32281	205	32372	213	32463	227
32191	92	32282	119	32373	205	32464	45
32192	25	32283	0	32374	122	32465	225
32193	34	32284	254	32375	126	32466	205
32194	69	32285	34	32376	24	32467	21
32195	92	32286	40	32377	50	32468	22
32196	175	32287	3	32378	42	32469	42
32197	50	32288	35	32379	83	32470	91
32198	71	32289	24	32380	92	32471	92
32199	92	32290	7	32381	237	32472	34
32200	231	32291	205	32382	91	32473	93
32201	253	32292	119	32383	78	32474	92
32202	52	32293	0	32384	127	32475	42
32203	13	32294	254	32385	126	32476	176
32204	254	32295	34	32386	230	32477	92
32205	58	32296	32	32387	192	32478	70
32206	40	32297	249	32388	32	32479	43
32207	248	32298	205	32389	11	32480	78
32208	254	32299	119	32390	126	32481	235
32209	13	32300	0	32391	184	32482	42
32210	202	32301	254	32392	56	32483	93
32211	10	32302	202	32393	16	32484	92
32212	127	32303	194	32394	192	32485	9
32213	254	32304	250	32395	35	32486	193
32214	234	32305	126	32396	126	32487	167
32215	202	32306	24	32397	185	32488	237
32216	66	32307	0	32398	56	32489	66
32217	127	32308	205	32399	11	32490	235
32218	254	32309	119	32400	201	32491	115
32219	225	32310	0	32401	33	32492	35
32220	40	32311	229	32402	240	32493	114
32221	27	32312	209	32403	216	32494	193
32222	254	32313	254	32404	25	32495	42

Grendel, il vostro renumber

Seguito figura 2.

32496	93	32522	35	32548	235	32574	71
32497	92	32523	126	32549	6	32575	92
32498	35	32524	230	32550	0	32576	207
32499	35	32525	192	32551	79	32577	255
32500	35	32526	202	32552	9	32578	42
32501	113	32527	176	32553	235	32579	176
32502	35	32528	125	32554	126	32580	92
32503	112	32529	42	32555	230	32581	70
32504	35	32530	83	32556	192	32582	43
32505	35	32531	92	32557	32	32583	78
32506	34	32532	58	32558	3	32584	9
32507	93	32533	76	32559	121	32585	35
32508	92	32534	127	32560	24	32586	24
32509	14	32535	237	32561	233	32587	190
32510	0	32536	91	32562	34	32588	10
32511	126	32537	78	32563	76	32589	0
32512	22	32538	127	32564	127	32590	10
32513	1	32539	114	32565	33	32591	0
32514	205	32540	35	32566	0	32592	201
32515	154	32541	115	32567	0	32593	0
32516	25	32542	35	32568	34	32594	0
32517	254	32543	78	32569	69	32595	0
32518	13	32544	35	32570	92	32596	0
32519	194	32545	70	32571	62	32597	0
32520	200	32546	35	32572	1	32598	0
32521	125	32547	9	32573	50	32599	0

300) non è l'equivalente di Print Fnd (300, 1100).

Spesso, durante l'attività preferita di un programmatore BASIC, si rende necessaria la renumerazione di un listato, sia perchè i numeri di linea sono talmente adiacenti da non permettere nuove inserzioni o aggiunte, sia per questioni di pura estetica (i listati stampati su una rivista è meglio che siano molto ordinati...).

Non è peraltro difficile procurarsi un semplice programmino, anche scritto in BASIC, che modifichi a nostro piacimento i numeri di linea: d'altra parte è assolutamente chiaro che se abbiamo per esempio un programma numerato con linee di incremento 10 e lo renumeriamo con base 1000 e incremento uno, il significato di un qualsiasi Goto all'interno del programma originale ne è stravolto, poichè cambia l'intero sistema di riferimento.

Per chiarire meglio l'idea è bene osservare la figura 3, che raffigura il programma da renumerare; come si nota, il passo delle linee è irregolare, e se diamo Run otteniamo una serie di Beep.

Renumerandolo in base 100 e passo dieci utilizzando un normale programma, si ottiene ciò che è illustrato dalla figura 4, e dando Run non otteniamo apparentemente nulla, se non un probabile 4-Out Of Memory Error dopo qualche secondo.

Sarebbe necessario utilizzare un renumber che "esplori" il programma BASIC alla ricerca delle istruzioni Goto, Gosub, List, Run, Llist, Restore e Line e ne modifichi i parametri rispettando la variazione di base e incremento.

Tale operazione è preclusa a qualsiasi programma BASIC, poichè se per esempio un List 2 deve diventare List 14, si rende necessario lo spostamento verso indirizzi maggiori di tutto il programma, per fare spazio alla cifra aggiuntiva ("2" occupa sette byte, mentre "14" ne occupa 8) e ciò è impossibile per un BASIC.

E' quindi per questo motivo che Grendel è stato scritto in linguaggio macchina: dando un Print Fn r (100, 10) si può ottenere lo stesso listato di figura 5 che è l'esatto equivalente dell'originale.

Come poi si nota dalla figura 6, Grendel corregge solo i valori numerici fissi e non le espressioni: occorre quindi prestare una particolare attenzione. Inoltre, se per qualsiasi motivo il programma non può essere renumerato con i parametri passati con la Fn, essi vengono sostituiti con i valori 1,1 avendo cioè base uno e stesso incremento.

Ciò potrebbe verificarsi a causa di numeri che supererebbero il 9999 o per un Out Of Memory imminente.

Figura 3

```
1REM prova
3 GO SUB 19
7 GO TO 3
19 BEEP 1,12
23 RETURN
```

Figura 4

```
100 REM prova
110 GO SUB 19
120 GO TO 3
130 BEEP 1,12
140 RETURN
```

Figura 5

```
100 REM prova
110 GO SUB 130
120 GO TO 110
130 BEEP 1,12
140 RETURN
```

Figure 3-4-5. *Questi tre listati dimostrano come sia necessario modificare un programma BASIC che deve essere renumerato, non solo con i numeri di linea iniziali, ma anche all'interno delle linee.*



Grendel, il vostro renumber

Accettabili

GO TO 10
GO SUB 50
LIST 100
LIST#3;100
LLIST#2;1000
RUN 30
RESTORE 1180
SAVE "Grendel" LINE 1
LLIST 45

Non accettabili

GO TO a
GO SUB 10-3
LIST 100-a
RESTORE k★10
RUN h+y/d
LLIST#2;d-f
SAVE "x" LINE VAL a$

Figura 6. *Possono essere modificati da Grendel solo i numeri singoli (accettabili) e non le espressioni numeriche.*

(1) calcolo dei parametri della Fn.
(2) salto all'indirizzo specificato da Usr, nella Def Fn.
(3) ritorno dalla subroutine.
(4) assegnazione di Bc sul canale desiderato.
(5) nuovi comandi.

Figura 7. *Queste sono le operazioni che il microprocessore espleta per eseguire un programma in linguaggio macchina con il sistema delle Def Fn.*

Note e conclusione

Poichè i parametri da utilizzare, con entrambi i programmi, sono trasferiti con l'ormai collaudatissimo metodo delle Fn, occorre sempre che in qualche parte del programma BASIC siano presenti le due Def Fnd (...) e Def Fn r(...).

Qualche lettore particolarmente acuto potrebbe inoltre notare che anche se si esegue una Print (Print Fn...) non viene stampato assolutamente niente. Se qualche ardimentoso non si dà per vinto e prova digitare Let 1=Fn r (...) e poi Print 1 riceve un inspiegabile 2 Variable Not Found: ciò è dovuto ad una particolare tecnica di ritorno al BASIC che permette di evitare al valore dei registri Bc di venire assegnati alla Usr e in definitiva alla Fn, ed è utilizzabile da qualsiasi routine linguaggio macchina che non necessiti di restituire valori numerici al BASIC, e sia da eseguire solo in modo diretto.

Invece di ritornare semplicemente con l'istruzione Ret, è possibile causare un errore chiamando la routine apposita, dopo aver compiuto il lavoro richiesto, e decidere il codice dell'errore (nel nostro caso OK ha codice 255).

Quindi, riferendosi al diagramma di figura 7, se interrompiamo lo svolgersi delle azioni al punto (3) cioè di ritorno dalla subroutine con:

```
RST 8
DEFB codice–1
```

il valore dei registri Bc non è depositato da nessuna parte e se il comando di chiamata era stato eseguito da programma BASIC, questo si fermerà con il codice di errore definito dalla direttiva Assembly Defb. ■

PERSONAL SOFTWARE

Gollum per C 64

Parte seconda

Concludiamo con i commenti al programma

di *Stefano Guarinelli*

In questa seconda e conclusiva puntata viene presentato il listato del programma con il commento delle varie istruzioni.
Nel seguire il commento di ogni istruzione, o blocchi di istruzioni, all'interno dei sottoprogrammi è opportuno fare riferimento alla figura 1, (in fondo all'articolo) che ne evidenzia i livelli di chiamata.
Nel seguito, quando si farà riferimento a un sottoprogramma lo si indicherà con il numero della prima istruzione dello stesso. Se invece si farà riferimento specifico ad una istruzione, o blocco di istruzioni, il numero o l'intervallo in questione verrà preceduto dall'abbreviazione "istr."

REMarks

20 - La variabile PP viene posta a 1, non appena si è giocata la prima partita, e fa saltare la presentazione iniziale del programma in tutte le partite successive (vedi 60000).
40 - Viene identificato il tipo di calcolatore su cui è stato caricato il programma (Commodore 4032 o C 64), (vedi 63900).
45-80 - Presentazione del programma. L'istruzione di linea 45 fa saltare le Data che si riferiscono ai gradi di libertà (vedi 33000), per partire invece da quelle che contengono le posizioni dello schermo e i codici ASCII dei simboli grafici utilizzati. Il valore –1 serve solo da separatore ed è del tutto arbitrario. Nella presentazione, alla variabile A viene sommato il registro base dello schermo RB, fissato dal 63900.
110-150 - Data relativi ai gradi di libertà (vedi 33000).
160-360 - Data relativi alla presentazione iniziale.
365-550 - Proseguimento della presentazione.
600 - La ulteriore chiamata al 63900 serve solo nel caso si tratti di partite successive alla prima, dove la istr. 40 non viene considerata.
605 - Vengono inizializzate tutte le variabili grafiche (vedi 47000).
610-780 - Dimensionamento di array e matrici. SC, SS, SA e DR erano già state esaminate nella prima parte dell'articolo.
DC ha lo stesso comportamento di DR, con la sola differenza che assume valore non nullo anche per le gallerie cosiddette rocciose (dove non c'è traccia e quindi DR(X,Y)=0). Si vedrà in seguito il suo scopo (vedi 20000).
PG$ e PJ$ memorizzano le mosse del giocatore e di Gollum, per costruire il riepilogo al termine della partita (vedi 15000). Memorizzano le varie posizioni come stringhe di caratteri e non come numeri, con il solo scopo di risparmiare memoria (altrimenti avrebbero richiesto una struttura a matrice 50x50) (vedi istruzioni 6600-6680; istruzioni

Listato 1. *Il programma Gollum.*

```
FILE: GOLLUM.LST  PAG. 1
1 REM          ****************
2 REM          **** GOLLUM ****
3 REM          ** V38-221284 **
4 REM          ****************
5 REM
20 IF PP=1 THEN RUN 600
30 PRINT"{CLR}"
40 GOSUB 63900
45 READ A:IF A<>-1 THEN 45
50 FOR J=1 TO 138
60 READ A,B:A=A+RB
70 POKE A,B
75 GOSUB 40000
80 NEXT
110 DATA 0,1,1.5,2,2,2,1.5,1,0
120 DATA 1,1.5,2,2.5,3,2.5,2,1.5,1
130 DATA 1,2,2.5,3,3,3,2.5,2,1
140 DATA 1,1.5,2,2.5,3,2.5,2,1.5,1
150 DATA 0,1,1.5,2,2,2,1.5,1,0
160 DATA -1,368, 77, 392, 160, 428, 95, 3
    07, 78, 496, 223, 505, 105, 465, 160
170 DATA 327, 223, 287, 78, 430, 233, 460
    , 78, 412, 160, 344, 101, 285, 78
180 DATA 372, 160, 314, 101, 251, 100, 33
    6, 101, 446, 99, 340, 101, 335, 78
190 DATA 387, 160, 303, 78, 394, 101, 434
    , 101, 432, 160, 374, 160, 416, 76
200 DATA 459, 78, 486, 160, 472, 160, 468
    , 78, 507, 105, 489, 95, 349, 77
210 DATA 406, 233, 425, 160, 452, 160, 29
    8, 78, 330, 160, 292, 78, 376, 101
220 DATA 420, 76, 447, 160, 454, 160, 346
    , 78, 326, 160, 299, 77, 293, 77
230 DATA 288, 77, 294, 78, 290, 78, 414,
    160, 422, 160, 453, 78, 445, 78
240 DATA 455, 78, 450, 78, 334, 223, 500,
     223, 364, 160, 347, 223, 247, 100
250 DATA 448, 78, 463, 78, 462, 160, 411,
     76, 385, 160, 391, 233, 305, 78
260 DATA 324, 233, 308, 77, 365, 103, 388
    , 223, 404, 160, 429, 223, 369, 160
270 DATA 354, 101, 431, 105, 449, 160, 38
    4, 101, 325, 160, 329, 233, 338, 223
280 DATA 497, 78, 504, 160, 306, 77, 328,
     78, 502, 95, 456, 78, 252, 100
```

PERSONAL SOFTWARE

Gollum per C 64

Seguito listato Gollum.

```
300 DATA 371, 101, 380, 101, 427, 160, 30
    4, 101, 295, 77, 495, 160, 490, 160
310 DATA 485, 160, 470, 105, 498, 95, 246
    , 100, 342, 233, 487, 105, 474, 101
320 DATA 407, 223, 457, 77, 458, 160, 345
    , 223, 503, 160, 512, 95, 484, 95
330 DATA 313, 78, 444, 160, 499, 160, 501
    , 78, 466, 78, 331, 160, 333, 78
340 DATA 352, 233, 367, 78, 491, 160, 492
    , 105, 494, 95, 469, 95, 332, 223
350 DATA 378, 160, 382, 160, 390, 77, 408
    , 78, 424, 76, 418, 160, 409, 160
360 DATA 405, 103, 461, 77, 467, 160, 339
    , 78, 513, 78
365 AR=700:GOSUB 40000
370 AU$="S.GUARINELLI{ 2 SPAZI}-
    { 2 SPAZI}PERSONAL SOFTWARE
    { 2 SPAZI}1985"
380 LU=LEN(AU$)
390 C9=0:C8=39:R9=5:R8=14
400 FOR J=1 TO 40
410 POKERB+C9+(40*R9),64
420 POKERB+C8+(40*R8),64
430 C9=C9+1:C8=C8-1
450 NEXT
460 C9=0:C8=39:R9=17:R8=21
465 KC=KC+1
470 IF KC<2 THEN 400
480 PRINT"{ 18 GIU'}";
490 FOR J=1 TO LU
500 PRINTLEFT$(AU$,1);:AR=60:GOSUB40000
520 AU$=RIGHT$(AU$,LU-J)
540 NEXT
550 PRINT"{HOME}{ 23 GIU'}PER COMINCIARE
    BATTERE {RVS} RETURN {OFF}"
560 GET IN$:IF IN$=CHR$(13) THEN 600
570 RA=RND(1):GOTO 560
600 GOSUB 63900
605 GOSUB 47000
610 DIM SC(9,5)  :REM TRACCE E NASCOND.
640 DIM SS(9,5)  :REM RIQUADRI SPECIALI
650 DIM SA(10,6):REM RIQUADRI APERTI
700 DIM DR(9,5)  :REM DIREZIONE TRACCE
710 DIM DC(9,5)  :REM DIR.COMPLEMENTARE
720 DIM PG$(51)  :REM PERCORSO GOLLUM
740 DIM PJ$(51)  :REM PERCORSO UTENTE
750 DIM MP(9)    :REM MOSSE POSSIBILI
760 DIM AV(9,5)  :REM TRACCE CONFUSE
770 DIM GL(9,5)  :REM GRADI LIBERTA'
780 DIM IP(9,5)  :REM BOOL.DIS.RIQUADRI
805 PRINT"{CLR}{ 4 GIU'}{ 4 DES}IN 50 MOS
    SE DEVI TROVARE GOLLUM "
806 PRINT"{ 2 GIU'}{ 12 DES}BUONA FORTUNA
    !!!"
807 GOSUB 33000
810 PRINT"{ 11 GIU'}{ 5 DES}VUOI LE ISTRU
    ZIONI ? ";
811 GOSUB 50500
812 IF US$="N" THEN 815
814 GOSUB 31000
815 PRINT"{CLR}"
820 FOR T=39 TO 0 STEP-1
830 POKE RB+T,160:AR=1:GOSUB 40000
840 NEXT
850 FOR T=1 TO 21
860 POKE RB+(40*T),160:AR=1:GOSUB 40000
870 NEXT
900 PRINT"{HOME}{RVS}{ 6 SPAZI}A
    { 3 SPAZI}B{ 3 SPAZI}C{ 3 SPAZI}D
    { 3 SPAZI}E{ 3 SPAZI}F{ 3 SPAZI}G
    { 3 SPAZI}H{ 3 SPAZI}I ";
950 PRINT"{HOME}{ 3 GIU'}{RVS}1{ 4 GIU'}
    {SIN}2{ 4 GIU'}{SIN}3{ 4 GIU'}{SIN}4
    { 4 GIU'}{SIN}5"
970 PRINT"{HOME}"
980 GOSUB 63000
1100 FOR Y=1 TO 50
1180 CL=INT(RND(1)*9)+1
1200 RW=INT(RND(1)*5)+1
1210 IF IP(CL,RW)=1 THEN 1250
1220 GOSUB 51000
1230 PRINTRN$
1240 IP(CL,RW)=1
1250 NEXT
1300 GOSUB 62000
2000 GOSUB 45000
5000 REM SCELTA GIOCATORE
5050 PRINT"{HOME}{ 23 GIU'}{OFF}PUNTO DI
     PARTENZA:"
5250 GET RW$:IF RW$="" THEN 5250
5270 IF ASC(RW$)<49 OR ASC(RW$)>53 THEN 5
     250
5300 PRINT"{HOME}{ 23 GIU'}{ 18 DES}{RVS}
     "RW$" {OFF}"
5400 GET CL$:IF CL$="" THEN 5400
5450 IF ASC(CL$)<65 OR ASC(CL$)>73 THEN 5
     400
5460 RW=VAL(RW$)
5470 CL=ASC(CL$)-64
5500 PRINT"{HOME}{ 23 GIU'}{ 21 DES}{RVS}
     - "CL$" {OFF}"
5520 IF AA=0 THEN GOSUB 57000:GOSUB 55000
     :LC=CL:WR=RW:VR$=TM$:GOTO 5700
5530 LC=CL:WR=RW:VR$=TM$
5550 AA=1
5570 GOSUB 43000
5580 GOSUB 57000
5590 GOSUB 51000
5600 GOSUB 55000
5620 C1=CL:R1=RW
5630 CL=LC:RW=WR
5650 GOSUB 51000
5660 IF C1=LC AND R1=WR THEN 5680
5670 IF SA(CL,RW)=1 THEN PRINTVR$
5680 CL=C1:RW=R1
5700 PRINT"{HOME}{RVS}{ 40 SPAZI}";
5705 PRINT"{HOME}{ 3 GIU'}{RVS}
     { 4 GIU'}{SIN} { 4 GIU'}{SIN}
     { 4 GIU'}{SIN} { 4 GIU'}{SIN} "
5710 MS=MS+1
5715 IF MS>=50 THEN EX=1:GOTO 6600
5720 SA(CL,RW)=1
5750 PRINT"{HOME}{ 23 GIU'}{ 15 SPAZI}NUM
     ERO MOSSE:{ 4 SPAZI}";MS
5800 IF MS/5<>INT(MS/5) THEN 6000
5805 IF MS/20=INT(MS/20) THEN 6000
5810 MS$=STR$(MS)
5820 FOR J=1 TO 6
5830 CC=1
5840 PRINT"{HOME}{ 23 GIU'}{ 32 DES}{RVS}
     "MS$" {OFF}"
5850 AR=200:GOSUB 40000
5860 PRINT"{HOME}{ 23 GIU'}{ 32 DES}"MS$"
     "
5870 AR=200:GOSUB 40000
5880 NEXT J
5885 CK=CL:RK=RW
5887 IF NA>=3 THEN NT=3:GOTO 5890
5888 NT=NA
5890 FOR K=1 TO NT
5900 CL=INT(RND(1)*9)+1
5920 RW=INT(RND(1)*5)+1
5930 IF SA(CL,RW)=0 THEN 5900
```

Gollum per C 64

28000-28100; istruzioni 16100-16150; istruzioni 16300-16350).
MP precisa per ogni mossa possibile la relativa convenienza.
AV è una matrice di booleani. Quando Gollum lascia una traccia, se il riquadro corrispondente di AV vale 1, la traccia è confusa, se vale 0, non lo è.
GL memorizza i gradi di libertà di ogni riquadro.
IP è booleana e serve per la presentazione iniziale. Rende un po' più veloce il disegno random delle gallerie, escludendo le posizioni uguali a 1 le quali indicano che il riquadro è già stato disegnato.
2000 - Gollum si nasconde! (vedi 45000).
5000 - Inizia la parte di programma riservata all'interazione con l'utente.
5050-5450 - Vengono introdotte la riga e la colonna della casella scelta come punto di partenza. I test 5270 e 5450 servono a far "passare" solo i caratteri 1-5 per le righe e A-I per le colonne.
5460-5470 - Vengono trasformate in numeri le variabili introdotte come stringhe per le righe e le colonne.
5520 - La variabile booleana AA posta a 0 precisa che si tratta della scelta del punto di partenza. Viene aperta la galleria scelta (vedi 57000; vedi 55000) e vengono memorizzati gli indici, riga-colonna, della galleria stessa in LC e WR, nonché il tipo di galleria in VR$ (vedi 47000; vedi 41000). Si vedrà in seguito l'uso di queste tre variabili.
5530 - Viene effettuata la stessa operazione vista in 5520, in modo da "trattenere" i parametri fondamentali della galleria aperta precedentemente, prima di aprirne una nuova.
5550 - La variabile booleana AA posta a 1 precisa che la scelta del punto di partenza è stata effettuata e che quindi si passa al controllo degli spostamenti mediante il keypad numerico.
5570 - Viene richiesto lo spostamento del giocatore, mediante il keypad numerico, e visualizzato (vedi 43000).
5580-5600 - Viene aperta la galleria su cui il giocatore si è fermato (vedi 57000, 51000, 55000).
5620-5680 - Non appena il giocatore si sposta da una galleria ad un'altra, quella "vecchia" viene ridisegnata con lo stesso simbolo grafico, ma in reverse. Questo affinché il giocatore sappia sempre qual è la sua posizione (a meno di una chiusura della casella su cui si trova nel caso di una mossa multipla di 5 o 20). Questo set di istruzioni realizza quanto detto a parole. Da qui, inoltre, si capisce il significato delle istruzioni viste alle linee 5520 e 5530. Le variabili Cl e R1 sono temporanee e servono a conservare il valore della posizione corrente del giocatore. Il sottoprogramma 51000 infatti riconosce solo, come parametri, le variabili CL e RW; pertanto gli indici (riga-colonna) della vecchia galleria, contenuti in WR e LC, devono essere temporaneamente trasferiti in RW e CL, rispettivamente, e il contenuto di queste ultime deve essere precedentemente salvato e quindi ripristinato (vedi istruzione 5680). Quello del passaggio dei parametri per i sottoprogrammi è uno dei difetti più evidenti del linguaggio BASIC. La variabile VR$ contiene il disegno della galleria tale e quale a prima, ma in reverse (vedi 47000, 41000, 55000). Il test 5660 significa che se il giocatore non si è mosso, la vecchia galleria coincide con la nuova e quindi la VR$ non viene stampata. Nella istruzione 5670 si precisa invece che la VR$ viene disegnata solo se la vecchia galleria è una galleria aperta.
5700-5705 - Vengono cancellati dai bordi dello schermo le lettere e i numeri delle coordinate.
5710-5715 - La variabile MS conta il numero delle mosse realizzate dal giocatore. La variabile booleana EX viene posta a 1 quando Gollum non è stato trovato, essendo terminate le 50 mosse a disposizione.
5720 - La posizione degli indici CL e RW, corrispondente alla galleria dove si trova ora il giocatore, viene posta uguale a 1 in SA, per indicare che la galleria è aperta.
5800 - Si verifica se la mossa è un numero multiplo di 5; se lo è, si passa alla istruzione 5810.
5805 - Si verifica se la mossa è un numero multiplo di 20; se lo è, si passa alla istruzione 6000.
5810-5880 - Viene fatto lampeggiare (a intermittenza reverse) per sei volte il numero della mossa (vedi 40000).
5885 - Tutte le volte che viene usato il sottoprogramma 51000, come si è già visto per le istruzioni 5620-5680, i contenuti di CL e RW devono essere salvati (in questo caso da CK e RK) e, al termine, ripristinati. Questa operazione non verrà ulteriormente commentata.
5887-5970 - Procedura per la chiusura di 3 gallerie scelte a caso tra quelle aperte. La variabile NA conta il numero totale delle gallerie aperte. La variabile NT evita che il programma, come si dice in gergo, possa andare "in loop": questo potrebbe verificarsi nel caso in cui le gallerie aperte, ad esempio, fossero solo 2.

In questo caso il calcolatore continuerebbe a cercare una galleria aperta, da chiudere, che non esiste. Se invece le gallerie aperte sono in numero inferiore a 3, il totale viene fissato da NT.
La variabile RN$ contiene il disegno della galleria chiusa: tutte le parti grafiche relative alle gallerie sono considerate variabili di tipo stringa (vedi 47000). Il sottoprogramma 51000 serve per posizionare il cursore nel punto dello schermo dove deve essere disegnata la variabile grafica RN$ (vedi 51000).
6000 - Inizia la procedura per la chiusura di tutte le gallerie aperte, se la mossa è un numero multiplo di 20.
6100-6260 - Viene fatto lampeggiare (a intermittenza reverse) il numero della mossa (vedi 40000).
6300-6390 - Vengono chiuse tutte le gallerie dal punto di vista di ciò che "vede" il computer e cioè azzerando tutte le posizioni della già vista ma-



Gollum per C 64

Seguito listato Gollum.

```
5940 GOSUB 51000
5945 PRINTRN$;
5950 SA(CL,RW)=0
5960 NEXT K
5970 NA=NA-3:IF NA<0 THEN NA=0
6000 IF MS/20<>INT(MS/20) THEN 6500
6050 IF MS=0 THEN 6500
6100 MS$=STR$(MS)
6210 FOR P=1 TO 6
6220 PRINT"{HOME}{ 23 GIU'}{ 28 DES}
     { 3 V}{RVS}"MS$" {OFF}{ 3 V}"
6230 AR=200:GOSUB 40000
6240 PRINT"{HOME}{ 23 GIU'}{ 28 DES}
     { 3 SPAZI}"MS$"{ 4 SPAZI}"
6250 AR=200:GOSUB 40000
6260 NEXT P
6300 FOR P=1 TO 9
6320 FOR L=1 TO 5
6340 SA(P,L)=0
6360 NEXT L
6380 NEXT P
6390 NA=0
6400 GOSUB 62000
6500 IF AA=1 AND CC=1 THEN CL=CK:RW=RK
6510 CC=0
6550 GOSUB 20000
6600 MC$=STR$(CL):MR$=STR$(RW)
6650 PJ$(HU)=MC$+MR$
6680 HU=HU+1
6700 IF EX<>1 THEN 7000
6800 GOSUB 30000
6850 GOTO 60000
7000 GOTO 5530
15000 REM RIEPILOGO PERCORSO
15100 FU$="{GIU'}{DES}{OFF}[<A>][<S>]
      { 2 SIN}{GIU'}[<Z>][<X>]{OFF}"
15120 KH$="{GIU'}{DES}{OFF}{ 2 SPAZI}
      { 2 SIN}{GIU'}{ 2 SPAZI}"
15130 LA$="O[< 2 T>]P{ 4 SIN}{GIU'}[<G>]
      { 2 SPAZI}[<M>]T{ 4 SIN}{GIU'}[<G>]
      { 2 SPAZI}[<M>]{ 4 SIN}{GIU'}L
      [< 2 @>]@{ 3 SU}"
15150 FG$="{GIU'}{DES}{RVS}£[<*>]
      { 2 SIN}{GIU'}{OFF}[<*>]£"
15160 FD$="M [<G>]N{ 4 SIN}{GIU'}[<@>]MN
      [<@>]T{ 4 SIN}{GIU'} NM { 4 SIN}
      {GIU'}N [<G>]M"
15170 MS=1
15180 PRINT"{CLR}"
15190 CL=1:RW=1:GC=1:GR=1:XC=9:XR=4
15200 FOR G=1 TO 5
15300 FOR T=1 TO XC:GOSUB 51000:PRINTLA$:
      CL=CL+GC:NEXT
15320 CL=CL+(-GC):RW=RW+(GR)
15350 IF G=5 THEN 15500
15400 FOR T=1 TO XR:GOSUB 51000:PRINTLA$:
      RW=RW+GR:NEXT
15420 CL=CL+(-GC):RW=RW+(-GR)
15450 XC=XC-1:XR=XR-1:GC=-GC:GR=-GR
15500 NEXT G
15750 FOR U=1 TO 5
15760 PRINT"{HOME}{ 21 GIU'}{ 4 DES}"KH$:
      AR=150:GOSUB 40000
15800 PRINT"{HOME}{ 21 GIU'}{ 4 DES}"FU$"
      =TU":GOSUB 40000
15810 NEXT U
15820 FOR U=1 TO 5
15825 PRINT"{HOME}{ 21 GIU'}{ 13 DES}"KH$
      :GOSUB 40000
15830 PRINT"{HOME}{ 21 GIU'}{ 13 DES}"FG$
      "=GOLLUM":GOSUB 40000
15835 NEXT U
15900 PRINT"{HOME}{ 4 DES}PIU' VELOCE =
      {RVS} + {OFF} , PIU' LENTO = {RVS}
      - {OFF} ";
15920 AR=1500:GOSUB 40000
15950 AR=700
16000 FOR F=0 TO HU-1
16010 PRINT"{HOME}{ 23 GIU'}{ 27 DES}MOSS
      A N."MS;
16020 IF EE=0 THEN 16100
16030 CL=JC:RW=JR
16040 GOSUB 51000:PRINTKH$
16100 CL=VAL(LEFT$(PJ$(F),2))
16150 RW=VAL(RIGHT$(PJ$(F),1))
16160 GOSUB 51000:PRINTFU$
```

trice SA e la variabile delle gallerie aperte NA.

6400 - Vengono disegnate le gallerie chiuse (vedi 62000).

6500 - Viene ripristinato il valore delle variabili CL e RW solo nel caso in cui la mossa non sia la prima (AA = 1) e sia invece multipla di 5 (CC = 1) nella cui procedura di chiusura delle gallerie è stato impiegato il già citato sottoprogramma 51000 (se AA = 0 e CC = 0, CK e RK valgono 0).

6550 - Viene ceduto il controllo a Gollum (vedi 20000).

6600-6680 - Viene memorizzata la mossa del giocatore nel già visto vettore PJ$. Dapprima vengono trasformate in stringhe di caratteri il valore della colonna e quello della riga dell'ultima posizione del giocatore: essendo CL e RW numeri inferiori al 10, la somma delle stringhe relative sarà una stringa di soli due caratteri. La variabile HU conta le mosse del giocatore.

6700-7000 - La variabile EX posta a 1 (vedi istruzione 5715) segnala che le mosse possibili sono terminate e che Gollum non è stato trovato. Il tutto viene segnalato con un messaggio all'utente (vedi 30000) e quindi si passa alle opzioni di fine partita (vedi 60000). Se invece EX vale 0, si torna alla istr. 5530 e cioè si chiede una nuova posizione al giocatore.

All'istruzione 7000 termina la prima parte del programma principale. I gruppi di istruzioni che seguono, quando non diversamente specificato, appartengono a sottoprogrammi interni al principale (a diversi livelli di annidamento).

15000 - Questo sottoprogramma permette di rivedere, se lo si desidera (vedi 60000), il percorso seguito da Gollum e dal giocatore.

15100-15160 - Inizializzazioni di variabili grafiche non contenute in 47000, ma impiegate solo per il riepilogo.

- FU$: fa disegnare un quadrato, che rappresenta di volta in volta lo spostamento del giocatore.
- KH$: ad ogni nuova posizione fa cancellare Gollum dal riquadro della posizione precedente.
- LA$: fa disegnare ogni singola casella, che rappresenta ogni galleria.
- FG$: fa disegnare un rombo reverse, che rappresenta di volta in volta lo spostamento di Gollum.
- FD$: nel caso di ritrovamento di Gollum, da parte del giocatore, fa disegnare la "collisione" fra i due, nel momento in cui il quadrato FU$

Gollum per C 64

si sovrappone al rombo FG$.
15170 - MS rappresenta il numero di mosse. E' la stessa variabile impiegata nel conteggio delle mosse durante il gioco. Si potrebbe logicamente supporre che, in questo modo, sia andato perduto il valore delle mosse impiegate per trovare Gollum: questo, invece, è conservato nel contatore HU, che scandisce il vettore PJ$, (oppure in HG, che scandisce PG$).
15190-15500 - Viene disegnata la mappa delle gallerie come all'inizio della partita, ma con due differenze. Intanto ora le gallerie vengono disegnate da LA$ e non da RN$ (che è praticamente il suo reverse). Inoltre le varie LA$ si compongono a spirale partendo dall'angolo superiore sinistro dello schermo. Il breve algoritmo per la composizione della spirale è un "classico" nella didattica dei linguaggi di programmazione, pertanto ne daremo solo un breve accenno. La spirale viene costruita ripetendo cinque volte (istruzione 15200) la sequenza di riquadri LA$, dapprima orizzontalmente (istruzione 15300) e quindi verticalmente (istruzione 15400). Al termine di ogni sequenza ne riprende una successiva, ma con un numero sempre inferiore di riquadri (XC per le colonne e XR per le righe) e "a marcia indietro".
L'inversione di marcia al termine di ogni sequenza si ottiene invertendo di segno la variabile che incrementa la posizione delle righe GR e delle colonne GC. Ciò si realizza semplicemente imponendo a queste variabili, al termine di ogni sequenza, il segno meno.
15750-15900 - Disegno e scrittura di messaggi che contengono alcune spiegazioni sui simboli che compaiono nel riepilogo.
15920-15950 - Il sottoprogramma 40000 rallenta in alcuni punti l'esecuzione del programma. Il tempo di pausa è fissato di volta in volta dalla variabile AR (vedi 40000).
16000 - Inizia il riepilogo del percorso per le HU mosse del giocatore.
16020-16040 - La variabile EE (booleana), posta a 0, precisa che la mossa è la prima e quindi la variabile KH$, che deve cancellare la vecchia posizione del giocatore, non serve ancora. Se invece EE vale 1, allora la vecchia posizione è data (vedi istruzione 16200) da JC e JR.
16100-16200 - Come si era già detto in principio, il vettore PJ$ è un array di stringhe. Il valore di riga e colonna deve pertanto essere estratto dapprima come lettera a destra o a sinistra, rispettivamente, della relativa posizione di PJ$ e quindi trasformato in numero. Una volta estratti i valori RW e CL, attraverso il sottoprogramma 51000 si disegna FU$, che corrisponde al simbolo del giocatore.
16230 - Attraverso il sottoprogramma 56500 è possibile modificare il valore della pausa nell'ambito del sottoprogramma 40000, e cioè la velocità del riepilogo del percorso.
16250-17000 - Procedimento identico a quanto visto per il giocatore, questa volta per il disegno del rombo che rappresenta Gollum. Il test 16255 serve a saltare l'ultima parte del riepilogo quando le mosse sono terminate e Gollum non è stato trovato. Infatti Gollum effettua solo 49 spostamenti e non 50 come il giocatore.
17010-17150 - Se Gollum non è stato trovato (TV=0), finisce il riepilogo. Altrimenti si fanno sovrapporre il rombo e il quadrato, stampando la FD$.
17300-17350 - Termina il sottoprogramma.
20000 - Inizia il sottoprogramma che fa ragionare Gollum.

Seguito listato Gollum.

```
16200 JC=CL:JR=RW
16230 GOSUB 56500:GOSUB 40000
16250 IF EE=0 THEN 16300
16255 IF F=HU-1 AND TV=0 THEN 17000
16260 CL=QC:RW=QR
16270 GOSUB 51000:PRINTKH$
16300 CL=VAL(LEFT$(PG$(F),2))
16350 RW=VAL(RIGHT$(PG$(F),1))
16500 GOSUB 51000:PRINTFG$
16600 QC=CL:QR=RW
16700 EE=1:MS=MS+1
16800 GOSUB 56500:GOSUB 40000
17000 NEXT
17010 IF TV=0 THEN 17300
17020 PRINT"{HOME}{ 23 GIU'}{ 27 DES}MOSS
      A N."MS;
17050 CL=JC:RW=JR:GOSUB 51000:PRINTKH$
17100 CL=QC:RW=QR:GOSUB 51000:PRINTFU$
17150 GOSUB 51000:PRINTFD$:GOTO 17300
17300 PRINT"{HOME}{ 4 DES}{ 4 SPAZI}PER U
      SCIRE BATTERE {RVS} RETURN {OFF}
      { 6 SPAZI}"
17350 GET US$:IF US$<>CHR$(13) THEN 17350
17500 RETURN
20000 REM RISPOSTA CALCOLATORE
20040 JJ=1:GOSUB 29000
20050 PRINT"{HOME}{ 23 GIU'}{OFF}
      { 12 SPAZI}"
20100 AV(YC,YR)=0
21000 U1=ABS(CL-NC):U2=ABS(RW-NR)
21050 PH=U1+U2:V=1
21070 IF PH>=8 THEN ZS=ZS+1
21200 ON V GOTO 21310,21360,21410,21460,2
      1510,21560,21610,21660,21710
21310 IF NC>1 AND NR<5 AND SA(NC-1,NR+1)=
      0 THEN IC=NC-1:IR=NR+1:GOTO 22000
21320 MP(V)=-20:GOTO 23500
21360 IF NR<5 AND SA(NC,NR+1)=0 THEN IC=N
      C:IR=NR+1:GOTO 22000
21370 MP(V)=-20:GOTO 23500
21410 IF NC<9 AND NR<5 AND SA(NC+1,NR+1)=
      0 THEN IC=NC+1:IR=NR+1:GOTO 22000
21420 MP(V)=-20:GOTO 23500
21460 IF NC>1 AND SA(NC-1,NR)=0 THEN IC=N
      C-1:IR=NR:GOTO 22000
21470 MP(V)=-20:GOTO 23500
21510 IC=NC:IR=NR:GOTO 22000
21520 MP(V)=-20:GOTO 23500
```



Gollum per C 64

Seguito listato Gollum.

```
21560 IF NC<9 AND SA(NC+1,NR)=0 THEN IC=N
      C+1:IR=NR:GOTO 22000
21570 MP(V)=-20:GOTO 23500
21610 IF NC>1 AND NR>1 AND SA(NC-1,NR-1)=
      0 THEN IC=NC-1:IR=NR-1:GOTO 22000
21620 MP(V)=-20:GOTO 23500
21660 IF NR>1 AND SA(NC,NR-1)=0 THEN IC=N
      C:IR=NR-1:GOTO 22000
21670 MP(V)=-20:GOTO 23500
21710 IF NC<9 AND NR>1 AND SA(NC+1,NR-1)=
      0 THEN IC=NC+1:IR=NR-1:GOTO 22000
21720 MP(V)=-20:GOTO 23500
22000 PV=ABS(CL-IC)+ABS(RW-IR)
22020 IF PV=0 THEN MP(V)=-20:GOTO 23500
22040 AK=INT(RND(1)*4)+1
22050 UC=ABS(CL-IC):UR=ABS(RW-IR)
22100 IF PV<>4 OR UC<>UR THEN 22200
22110 PM=1
22150 IF VD<VS THEN 22170
22155 PV=PV-6
22160 IF T2=1 AND AK=3 AND Z2<Z1 THEN PV=
      PV+7
22170 IF IX=3 THEN PV=PV-4
22180 GOTO 22350
22200 IF PV<>3 OR ABS(UC-UR)<>1 THEN 2225
      0
22205 PM=1
22210 IF VD<VS THEN 22230
22215 PV=PV-6
22220 IF T2=1 AND AK=3 AND Z2<Z1 THEN PV=
      PV+7
22230 IF IX=1 THEN PV=PV-4
22240 GOTO 22350
22250 IF PV<>2 OR UC=UR THEN 22300
22255 PM=1
22260 IF VD<VS THEN 22280
22265 PV=PV-6
22270 IF T2=1 AND AK=3 AND Z2<Z1 THEN PV=
      PV+7
22280 IF IX=2 THEN PV=PV-4
22290 GOTO 22350
22300 IF PV<>1 AND PV<>2 THEN 22350
22310 PM=1
22320 IF VS<VD THEN 22350
22330 PV=PV-6
22340 IF T2=1 AND AK=3 AND Z1<Z2 THEN PV=
      PV+7
22350 IF IM=V AND T1=1 AND INT(RND(1)*4)+
      1<>2 THEN PV=PV-2
22400 IF V=5 AND IM=V AND MF=1 THEN PV=PV
      -2:MF=0
22600 IF V=5 AND PM=1 AND T1=1 THEN PV=PV
      -20
23000 PV=PV+SS(IC,IR)
23050 PV=PV+GL(IC,IR)
23100 IF T1=1 THEN KS=INT(RND(1)*3):GOTO
      23200
23150 IF T1=0 THEN KS=INT(RND(1)*2)
23200 IF DC(IC,IR)<>0 AND PM=1 THEN PV=PV
      -4:GOTO 23300
23250 IF DC(IC,IR)<>0 THEN PV=PV-2
23300 IF PM=0 THEN PV=PV+KS
23320 IF PM=1 AND T1=1 THEN PV=PV+INT(KS/
      2)
23350 IF V=5 AND T1=1 THEN PV=PV-3
23400 IF V=10-IM AND T1=1 THEN PV=PV-2
23450 MP(V)=PV
23500 PM=0:V=V+1:IF V<10 THEN GOTO 21200
24000 IF ZS<5 THEN 26000
24020 ZP=ZP+1
24100 FOR V=1 TO 9
24110 IF V=5 THEN MP(V)=-20
24120 IF MP(V)<=-4 THEN 24140
24130 MP(V)=-MP(V)
24140 NEXT V
24150 IF ZP>3 THEN ZS=0:ZP=0
26000 V=1:MX=MP(V):IM=V
26100 FOR V=2 TO 9
26200 IF MP(V)>MX THEN MX=MP(V):IM=V
26300 NEXT V
26480 E=1
26500 FOR V=1 TO 9
26550 IF MP(V)=MX THEN MT(E)=V:E=E+1
26600 NEXT V
26650 IF E=2 THEN 26800
26700 RA=INT(RND(1)*E):IF RA=0 THEN 26700

26750 IM=MT(RA)
26800 MV=IM
26820 IF MV=5 THEN MF=MF+1
27500 GOSUB 35000
28000 MC$=STR$(NC):MR$=STR$(NR)
28050 PG$(HG)=MC$+MR$
28100 HG=HG+1
28500 PRINT"{HOME}{ 23 GIU'}{RVS} PUOI MU
      OVERE {OFF}"
28600 RETURN
29000 REM RICERCA MS DOPPIA PIU' FREQ.
29100 FM(1)=DF:FM(2)=DI:FM(3)=DN
29150 HM=FM(1):IX=1
29200 FOR T=2 TO 3
29250 IF HM>FM(T) THEN 29400
29300 HM=FM(T):IX=T
29400 NEXT T
29900 RETURN
30000 REM GOLLUM NON TROVATO
30010 BL$="{ 39 SPAZI}"
30020 PRINT"{HOME}{ 23 GIU'}"BL$
30050 FOR J=1 TO 8
30100 PRINT"{HOME}{ 23 GIU'}{ 14 DES}
      {RVS}***** 50 *****{OFF}"
30150 AR=200:GOSUB 40000
30200 PRINT"{HOME}{ 23 GIU'}{ 14 DES}****
      * 50 *****"
30250 AR=200:GOSUB 40000
30300 NEXT J
30500 PRINT"{CLR}{ 6 GIU'}{DES}LE TUE POS
      SIBILITA' DI TROVARLO SONO ":PRINT"
      {GIU'}{DES}TERMINATE."
30550 PRINT"{ 3 GIU'}{ 4 DES}GOLLUM E' RI
      USCITO A FUGGIRE !"
30600 AR=3000:GOSUB 40000
30800 RETURN
31000 REM ISTRUZIONI
31100 PRINT"{CLR}[< 40 I>]";
31110 PRINT"{RVS}{ 14 SPAZI}ISTRUZIONI
      { 16 SPAZI}";
31120 PRINT"{RVS}[< 40 I>]";
31150 PRINT:PRINT"TU HAI LA POSSIBILITA'
      DI SPOSTARTI DI "
31160 PRINT"UNO O DUE RIQUADRI PER OGNI M
      OSSA.":PRINT
31170 PRINT"GOLLUM INVECE PUO' SPOSTARSI
      DI UN SOLO":PRINT"RIQUADRO PER VOLT
      A.
31200 PRINT:PRINT"UNA VOLTA FISSATE LE CO
      ORDINATE{ 4 SPAZI}7 8 9";
31210 PRINT"DEL PUNTO DI PARTENZA (RIGA-
      { 8 SPAZI}MBN ";
31220 PRINT"COLONNA), USA I TASTI NUMERIC
      I{ 5 SPAZI}4C C6";
31230 PRINT"PER SPOSTARTI:{ 22 SPAZI}NBM
      "
```

Gollum per C 64

20040 - La variabile booleana JJ è di scarsa importanza ed ha scopi puramente grafici (vedi istruzioni 48100-48110). Il sottoprogramma 29000, attraverso la variabile IX, fornisce l'indice del tipo di mossa doppia più frequente del giocatore (vedi 29000).
20050 - Viene cancellato il messaggio "Puoi muovere".
20100 - Viene azzerata la posizione di AV, alla quale corrispondeva la mossa precedente, i cui indici di riga e colonna sono contenuti in YR e YC (vedi 35000), ad indicare l'eliminazione della eventuale traccia confusa.
21000-21050 - Vengono determinate le distanze per colonna e per riga fra la posizione di Gollum (NC,NR) e quella del giocatore (CL,RW). Vengono infine sommate nell'unica variabile PH. L'uso della variabile V, invece, sarà chiaro in seguito.
21070 - Se PH è maggiore o uguale a 8, significa che Gollum e il giocatore sono molto distanti. Se ciò avviene, parte il contatore ZS, in base al quale, giudicherà dell'eccessiva permanenza del giocatore in una determinata zona.
21200 - Inizia il ciclo per determinare la convenienza di ogni mossa possibile. La variabile V precisa il tipo di mossa ed ha lo stesso significato di quanto visto in figura 1, nella prima parte dell'articolo.
21310-21720 - Determinazione della pericolosità della posizione per ciò che riguarda la maggiore o minore distanza tra Gollum e il giocatore. I test che seguono sono tutti concettualmente uguali; pertanto ne esamineremo uno solo come esempio: istruzione 21410-21420. Si entra in questo test mediante il Computed-Goto di linea 21200, quando la V vale 3, e cioè quando Gollum si accinge a valutare se sia possibile, e con quale pericolosità, effettuare lo spostamento di una casella a Sud-Est. Ciò è possibile, se Gollum non si trova nella posizione estrema dei riquadri del bordo inferiore e di quello destro (NC<9 e NR<5) e se la galleria nella quale si accinge ad entrare (che ha indici NR+1 e NC+1) non è aperta. Se la possibilità esiste, vengono fissati due valori provvisori IC e IR, che contengono i nuovi valori di colonna e riga. Se la possibilità non esiste, viene fissata addirittura la convenienza della mossa 3 nella terza posizione del vettore mosse possibili, MP, dandole un valore convenzionale – 20, che verrà riconosciuto più avanti come quello di mossa non realizzabile.
22000 - Per ogni V viene fissato quindi un valore di pericolosità PV, determinato esattamente come PH in istruzione 21050, ma con le variabili provvisorie IC e IR di colonna e riga.
22020 - Questo test serve ad evitare che Gollum vada a nascondersi nella stessa galleria dove si trova il giocatore, qualora questa sia rimasta chiusa in seguito al raggiungimento di una mossa multipla di 5 o 20. La cosa sembra inutile, ma il computer non sa, se non glielo si dice, che in questo caso Gollum e il suo inseguitore potrebbero "vedersi". Per il calcolatore nulla è mai troppo assurdo o scontato, e la realizzazione di questo test si è resa necessaria proprio perché in una delle numerose partite di prova Gollum, in un eccesso di furbizia, si era messo ad inseguire lui il suo inseguitore e, come se non bastasse, aveva probabilmente pensato di ingannarlo bene, nascondendosi là dove il giocatore non l'avrebbe mai sicuramente cercato.
22040 - AK è uno dei parametri di imprevedibilità. Vedremo il suo impiego in seguito.
22050 - Vengono calcolate le differenze fra riga e riga, colonna e colonna di Gollum e giocatore. In base al loro valore, infatti, si potrà valutare non solo la distanza, ma anche il tipo di mossa che separa Gollum dal suo inseguitore.
22100-22340 - Inizia una serie di test che cerca di valutare la pericolosità della posizione che assumerebbe Gollum nel caso eseguisse la mossa V, secondo la distanza e il tipo di mossa che lo separerebbero dal giocatore.
22100-22180 - Questo gruppo di istruzioni riguarda l'eventualità che con la mossa V Gollum si vada a collocare in una posizione raggiungibile dal giocatore in una sola mossa doppia, secondo la diagonale di entrambe le caselle. In questo caso la variabile booleana PM, che precisa la massima pericolosità, viene posta a 1. Se si osserva che il giocatore predilige le mosse doppie (VD) a quelle singole (VS) (vedi 43000), il parametro PV, che specifica, come si è visto, la convenienza (o la minor pericolosità) di ogni mossa, viene decrementato di 6 unità. Se tuttavia si è osservato che il giocatore seguendo le tracce di Gollum (T2=1: vedi 55000) ha la tendenza invece a preferire le mosse singole (Z1) a quelle doppie (Z2), e se il fattore di imprevedibilità AK vale 3 (AK è un intero compreso fra 1 e 4), allora PV sale di 7 unità. Comunque, se di tutte le mosse doppie realizzate dal giocatore questa è la più frequente (vedi 29000), allora PV scende di 4 unità.
22200-22290 - Con meccanismo analogo a quanto visto per la posizione distante due caselle secondo la diagonale vengono valutate le posizioni distanti due caselle, raggiungibili secondo la combinazone diagonale-retta (o retta-diagonale) e retta-retta.
22300-22340 - Viene valutata la posizione distante una casella.
22350 - Se lo spostamento scelto come migliore da Gollum alla mossa precedente (cioè il suo spostamento effettivo precedente) coincide con quello che si sta valutando (V), e se il giocatore è sulle tracce di Gollum (per la differenza fra T1 e T2 si veda 55000), nonché se è diverso da 2 un certo fattore di imprevedibilità compreso fra 1 e 4, allora PV decresce di 2 unità. In sostanza, si cerca di far sì che, se seguito, Gollum se ne vada un po' a "zig-zag".
22400 - Si cerca di evitare che Gollum stia fermo troppo a lungo. MF vale 1, se la mossa precedente a IM (che è la precedente di quella di cui si

stanno valutando tutte le possibilità) era 5.
22600 - Si cerca di evitare, anzi direi che si vieta (vista l'entità del decremento di PV), a Gollum di fermarsi in situazioni di pericolo massimo (PM = 1) e di giocatore sulle sue tracce (T1 = 1).
23000 - Si somma a PV il tipo di galleria che si raggiungerebbe se la scelta della mossa fosse per lo spostamento V. Il tipo di galleria vale 0, se la galleria è a fondo terroso, 1, se a fondo roccioso. In questo modo si fa sì che Gollum abbia una certa predilezione per quelle gallerie dove le sue tracce non rimangono.
23050 - Si somma a PV il valore del grado di libertà del riquadro, della cui posizione si sta valutando la convenienza.
23100-23150 - Viene fissato un altro parametro di imprevedibilità KS in maniera differente, a seconda che il giocatore sia sulle tracce di Gollum (T1 = 1) oppure no.(T1 = 0). Nel primo caso KS è un numero compreso tra 0 e 2; nel secondo caso tra 0 e 1.
23200 - Quando la pericolosità è massima (PM = 1) si cerca di evitare che Gollum possa tornare su una casella per la quale è già passato non oltre quattro mosse prima. Si usa la matrice DC e non la DR come matrice di verifica, perché se Gollum fosse passato su una galleria a fondo roccioso, DR avrebbe, in corrispondenza, valore nullo e quindi non sarebbe possibile sapere del suo passaggio.
23300 - Se non ci si trova in situazione di pericolo massimo, a PV viene sommato il parametro di imprevedibilità KS.
23320 - Se il pericolo è massimo, invece, viene sommato un parametro di imprevedibilità ridotto.
23350 - Si cerca di evitare che Gollum si fermi (V = 5) quando il giocatore è sulle sue tracce (T1 = 1).
23400 - Si cerca di evitare che Gollum torni indietro rispetto alla mossa effettiva precedente, se il giocatore è sulle sue tracce (se per esempio la mossa precedente IM era stata la 3, per tornare indietro Gollum dovrebbe fare la 7 e cioè la 10-3).
23450-23500 - Il valore di PV così calcolato per lo spostamento V viene memorizzato nella V-esima posizione di MP. Quindi il ciclo iniziato alla 21200 si chiude e si passa a valutare un altro spostamento possibile, incrementando V.
24000-24020 - Se ZS vale 5, significa che il giocatore sta rimanendo per troppo tempo nella stessa zona. A questo punto Gollum per 3 mosse (mediante l'uso della variabile ZP) si muoverà scegliendo la mossa peggiore fra quelle che hanno convenienza superiore a – 4. Poiché la scelta della mossa più conveniente avviene per selezione di un massimo, la peggiore si ottiene imponendo il segno – a tutti i valori di MP superiori a – 4.
24100-24140 - Viene imposto, come si è detto sopra, il segno – alle convenienze delle singole mosse, contenute in MP, escluso la 5 (stai fermo) e quelle con convenienza inferiore a – 4.
24150 - Quando ZP supera il valore 3 si torna alla normale procedura.
26000-26300 - Viene cercata la convenienza più alta MX e l'indice IM della corrispondente posizione in MP.
26480-26600 - Vengono cercate eventuali convenienze uguali fra di loro e uguali a MX, in MP, riempiendo il vettore MT; il vettore MT non contiene le convenienze, bensì gli indici corrispondenti alle posizioni in MP, che, come si è visto, rappresentano gli spostamenti possibili.
26650 - Se E vale 2, significa che esiste un solo elemento massimo (uguale a se stesso; da qui E = 2) e quindi non vi sarà alcuna ulteriore selezione.
26700-26800 - Se invece le convenienze uguali sono più di una, allora ne verrà estratta una a caso mediante l'indice RA, che è un numero compreso tra 1 ed E-1. La RA-esima posizione di MT contiene quindi lo spostamento più conveniente, che viene memorizzato in MV.
26820 - Se la MV fa star fermo Gollum, allora viene incrementata la già vista variabile MF (vedi istruzione 22400).
27500 - Viene chiamato il sottoprogramma che effettua l'aggiornamento di tracce e riquadri (vedi 35000).
28000-28100 - Come per il giocatore, anche per Gollum viene memorizzato in PG$ lo spostamento realizzato, che servirà per il riepilogo finale.
28500-28600 - Viene stampato il messaggio "Puoi muovere" e quindi il sottoprogramma termina.
29000 - Questo sottoprogramma calcola quale fra le mosse doppie realizzate dal giocatore è la più frequente (diagonale-diagonale; retta-retta; retta-diagonale).
29100 - Vengono memorizzate nel vettore FM: la DF, che conta le mosse diagonale-retta (e viceversa) (vedi istruzione 43870); la DI, che conta le mosse retta-retta (vedi istruzione 43900) e la DN, che conta le mosse diagonale-diagonale (vedi istruzione 43890).
29150-29900 - Vengono ricercati il più alto valore memorizzato e quindi collocato in HM, e l'indice corrispondente in IX, che è la variabile utilizzata dal sottoprogramma 20000. Quindi il sottoprogramma termina.
30000 - Questo sottoprogramma fornisce una serie di messaggi nel caso Gollum non sia stato trovato.
30010-30300 - Viene fatto lampeggiare il numero 50, per indicare che le mosse a disposizione del giocatore sono terminate.
30500-30800 - Stampa di un messaggio e fine del sottoprogramma.
31000-32000 - Questo sottoprogramma fa comparire sul video, a richiesta, le istruzioni fondamentali del gioco.
33000 - Questo sottoprogramma provvede al riempimento della matrice GL, che contiene i gradi di libertà dei riquadri.
33010 - Viene inizializzato il puntatore alle Data, per la loro lettura.
33100-33600 - Viene riempito GL,

Seguito listato Gollum.

```
31260 PRINTSPC(35)"1 2 3";
31300 PRINT"{ 40 C}";
31320 PRINT"SIGNIFICATO DEI RIQUADRI APER
      TI:":PRINT
31330 PRINTRQ$:PRINT"{ 4 SU}{ 5 DES}GALLE
      RIA VUOTA A FONDO TERROSO."
31340 PRINT"{ 5 DES}GOLLUM NON E' PASSATO
      DI QUI."
31350 PRINT:PRINT"{ 8 DES}( CONTINUA ) -
      BATTERE {RVS} RETURN {OFF}"
31360 GET US$:IF US$<>CHR$(13) THEN 31360
31500 PRINT"{CLR}";:PRINTRS$:PRINT"
      { 4 SU}";
31520 PRINT"{ 5 DES}GALLERIA VUOTA MA A F
      ONDO ROCCIOSO"
31530 PRINT"{ 5 DES}SE GOLLUM E' PASSATO
      DI QUI LE SUE"
31540 PRINT"{ 5 DES}TRACCE NON SONO COMUN
      QUE RIMASTE.":PRINT:PRINT
31650 PRINTD7$:PRINT"{ 4 SU}";
31660 PRINT"{ 5 DES}GOLLUM E' PASSATO DI
      QUI ED E' "
31670 PRINT"{ 5 DES}ANDATO VERSO NORD-OVE
      ST.":PRINT:PRINT
31680 PRINT:PRINTD5$:PRINT"{ 4 SU}";
31690 PRINT"{ 5 DES}GOLLUM E' PASSATO DI
      QUI MA ESSENDO";
31700 PRINT"{ 5 DES}MOLTO VICINO HA LASCI
      ATO VOLUTAMEN-";
31720 PRINT"{ 5 DES}TE TRACCE CONFUSE."
31750 PRINT:PRINT
31760 PRINT"LE TRACCE CONFUSE RESTANO TAL
      I PER UNA "
31770 PRINT"MOSSA A PARTIRE DA QUANDO SON
      O STATE LA-";:PRINT"SCIATE."
31780 PRINT"OGNI TRACCIA SI CANCELLA DOPO
       QUATTRO{ 2 SPAZI}"
31790 PRINT"MOSSE A PARTIRE DA QUANDO E'
      STATA LA-":PRINT"SCIATA."
31800 PRINT:PRINT"PER USCIRE BATTERE
```

```
      {RVS} RETURN {OFF}"
31900 GET US$:IF US$<>CHR$(13)THEN 31900
32000 RETURN
33000 REM GRADI LIBERTA' RIQUADRI
33010 RESTORE
33100 FOR J=1 TO 5
33150 FOR T=1 TO 9
33200 READ A
33250 GL(T,J)=A
33300 NEXT T
33350 NEXT J
33600 RETURN
35000 REM AGGIORNAMENTO RIQUADRI
35050 YC=NC:YR=NR
35100 IF MV=8 THEN NR=NR-1
35150 IF MV=2 THEN NR=NR+1
35200 IF MV=4 THEN NC=NC-1
35250 IF MV=6 THEN NC=NC+1
35300 IF MV=1 THEN NC=NC-1:NR=NR+1
35350 IF MV=3 THEN NC=NC+1:NR=NR+1
35400 IF MV=7 THEN NC=NC-1:NR=NR-1
35410 IF MV=9 THEN NC=NC+1:NR=NR-1
35420 IF SC(OC(1),OW(1))<>1 THEN 35460
35430 SC(OC(1),OW(1))=0:DR(OC(1),OW(1))=0
      :DC(OC(1),OW(1))=0
35460 FOR P=1 TO 4
35470 OC(P)=OC(P+1):OW(P)=OW(P+1)
35475 SC(OC(P),OW(P))=P
35480 NEXT P
35500 SC(NC,NR)=5:OC(5)=NC:OW(5)=NR
35550 IF SS(YC,YR)=0 THEN DR(YC,YR)=MV
35560 DC(YC,YR)=MV
35650 RA=INT(RND(1)*2)+1
35700 IF RA=1 THEN AV(YC,YR)=1
37000 RETURN
40000 REM SUBR. ARRESTO VIDEO
40100 FOR C=1 TO AR:NEXT:RETURN
41000 REM RIQUADRI CON TRACCE
41300 IF AV(CL,RW)=1 THEN TN$=D5$:TM$="
      {RVS}"+D5$:CF=1:GOTO 42000
41450 IF DR(CL,RW)=1 THEN TN$=D1$:TM$=B1$
```

quindi il sottoprogramma termina.

35000 - Questo sottoprogramma provvede all'aggiornamento delle tracce e dei riquadri.

35050 - Vengono memorizzati in YC e YR la colonna e la riga della precedente posizione di Gollum.

35100-35410 - A seconda dello spostamento scelto da Gollum (MV) vengono determinati gli indici di colonna e riga della nuova posizione.

35420 - Gli array OC e OW contengono rispettivamente gli indici delle colonne e delle righe delle ultime 5 posizioni di Gollum. La prima posizione, di indice 1 quindi, contiene la mossa più "vecchia". Se la corrispondente posizione della matrice SC non vale 1, significa che Gollum è tornato in una casella per la quale era già passato precedentemente. L'eliminazione della traccia più vecchia, e quindi l'azzeramento di SC, porterebbe, di conseguenza, anche all'eliminazione di una traccia più recente. Il test 35420 impedisce che ciò avvenga.

35430 - Se invece la posizione in SC della mossa più vecchia vale 1, allora si provvede al suo azzeramento. Vengono inoltre azzerate le rispettive posizioni delle due matrici DR e DC per la gestione delle tracce e il passaggio di Gollum.

35460-35480 - Gli indici di colonne e righe in OC e OW vengono fatti scorrere di una posizione, in modo che il primo indice vada perduto e il quinto sia reso disponibile. La matrice SC, per ogni colonna e riga, assume il valore della variabile di conteggio P del ciclo For: vale cioè 4 per la mossa più recente e 1 per quella più vecchia, 3 e 2 per quelle intermedie.

35500 - Assume valore 5 in SC la nuova posizione di Gollum. Inoltre, nella quinta posizione di OC e OW, resasi disponibile, vengono collocati gli indici di colonna e riga della nuova galleria.

35550 - Il tipo di traccia lasciato da Gollum ha la stessa codifica degli spostamenti. La matrice DR assume il valore della traccia di tipo MV, in corrispondenza della nuova posizione di Gollum, solo se la galleria non è a fondo roccioso.

35560 - La matrice DC, invece, assume il valore MV anche se la galleria è a fondo roccioso.

35650-37000 - Se il numero RA (che può valere 1 oppure 2) vale 1, allora la traccia lasciata da Gollum è confusa e ciò lo si realizza ponendo a 1 la matrice di booleani AV. Quindi il sottoprogramma termina.

40000-40100 - Questo sottoprogramma crea pause di tempo diverse, a seconda del valore della variabile AR, fissata di volta in volta dal programma o sottoprogramma chiamante.

41000 - Questo sottoprogramma converte i codici della matrice DR

Gollum
per C 64

Seguito listato Gollum.

```
41500 IF DR(CL,RW)=2 THEN TN$=TB$:TM$="
      {RVS}"+TB$
41550 IF DR(CL,RW)=3 THEN TN$=D3$:TM$="
      {RVS}"+D3$
41600 IF DR(CL,RW)=4 THEN TN$=TS$:TM$="
      {RVS}"+TS$
41650 IF DR(CL,RW)=6 THEN TN$=TD$:TM$="
      {RVS}"+TD$
41700 IF DR(CL,RW)=7 THEN TN$=D7$:TM$="
      {RVS}"+D7$
41750 IF DR(CL,RW)=8 THEN TN$=TA$:TM$="
      {RVS}"+TA$
41800 IF DR(CL,RW)=9 THEN TN$=D9$:TM$=B9$

42000 RETURN
43000 REM PERCORSO GIOCATORE
43020 CM=0:DP=0:SG=0
43050 GS$="{GIU'}{DES}{RVS}UI{ 2 SIN}
      {GIU'}JK{OFF}"
43060 CH$="{GIU'}{DES}{RVS}{ 2 SPAZI}
      { 2 SIN}{GIU'}{ 2 SPAZI}{OFF}"
43120 GET PR$:IF PR$="" THEN 43120
43150 IF ASC(PR$)<13 THEN 43120
43200 IF ASC(PR$)>13 AND ASC(PR$)<49 THEN
       43120
43250 IF ASC(PR$)=53 OR ASC(PR$)>57 THEN
      43120
43300 IF PR$=CHR$(13) AND CM=0 THEN 43120

43310 IF PR$=CHR$(13) THEN VS=VS+1:SG=1
43320 IF PR$<>CHR$(13) AND CM<>0 THEN VD=
      VD+1:DP=1
43350 GOSUB 52000
43360 IF ME=1 THEN 43120
43400 GOSUB 51000
43450 CM=CM+1
43500 PRINTGS$
43510 IF CM=1 THEN CY=CL:RY=RW:VM=VAL(PR$
      )
43550 IF CM<2 THEN 43120
43700 CX=CL:RX=RW
43720 CL=CY:RW=RY
43750 GOSUB 51000
43800 IF SA(CL,RW)=0 THEN PRINTCH$:GOTO 4
      3850
43820 IF DR(CL,RW)<>0 THEN GOSUB 41000:PR
      INTTM$:GOTO 43850
43830 IF SS(CL,RW)=1 THEN PRINTR2$:GOTO 4
      3850
43840 IF SS(CL,RW)=0 THEN PRINT"{RVS}"RQ$

43850 CL=CX:RW=RX
43855 IF PR$=CHR$(13) THEN 44000
43860 NQ=VAL(PR$)
43870 IF NQ<>VM THEN DF=DF+1:GOTO 44000
43890 IF (NQ=VM) AND (NQ=1 OR NQ=3 OR NQ=
      7 OR NQ=9) THEN DN=DN+1:GOTO 44000
43900 IF NQ=VM THEN DI=DI+1
44000 RETURN
45000 REM DISP. RIQUADRI SPECIALI
45100 CL=INT(RND(1)*9)+1
45150 RW=INT(RND(1)*5)+1
45200 SC(CL,RW)=5:OC(5)=CL:OW(5)=RW
45220 NC=CL:NR=RW
45300 TD=INT(RND(1)*11)+5
45350 FOR N=1 TO TD
45400 CL=INT(RND(1)*9)+1
45450 RW=INT(RND(1)*5)+1
45500 SS(CL,RW)=1
45540 NEXT N
46000 RETURN
47000 REM INIZIALIZZAZIONE RIQUADRI
47100 TS$="M{ 2 SPAZI}N{ 4 SIN}{GIU'} <<
      { 4 SIN}{GIU'}[< 4 F>]{ 4 SIN}
      {GIU'}N{ 2 SPAZI}M"
47120 TD$="M{ 2 SPAZI}N{ 4 SIN}{GIU'} >>
      { 4 SIN}{GIU'}[< 4 D>]{ 4 SIN}
      {GIU'}N{ 2 SPAZI}M"
47140 TA$="M[<F>] N{ 4 SIN}{GIU'} [<F>]↑
      { 4 SIN}{GIU'} [<F>]↑ { 4 SIN}
      {GIU'}N[<F>] M"
47160 TB$="M[<F>] N{ 4 SIN}{GIU'} [<F>]V
      { 4 SIN}{GIU'} [<F>]V { 4 SIN}
      {GIU'}N[<F>] M"
47180 D7$="M{ 2 SPAZI}N{ 4 SIN}{GIU'}
      [<B>]↑ { 4 SIN}{GIU'} ←[<B>]
      { 4 SIN}{GIU'}N{ 2 SPAZI}M"
47200 D1$="M{ 2 SPAZI}N{ 4 SIN}{GIU'} <
      {RVS}[<B>]{OFF} { 4 SIN}{GIU'}
      {RVS}[<B>]{OFF}V { 4 SIN}{GIU'}N
      { 2 SPAZI}M"
47210 B1$="{RVS}M{ 2 SPAZI}N{ 4 SIN}
      {GIU'} <{OFF}[<B>]{RVS} { 4 SIN}
      {GIU'} {OFF}[<B>]{RVS}V { 4 SIN}
      {GIU'}N{ 2 SPAZI}M"
47220 D9$="M{ 2 SPAZI}N{ 4 SIN}{GIU'} >
      {RVS}[<B>]{OFF} { 4 SIN}{GIU'}
      {RVS}[<B>]{OFF}↑ { 4 SIN}{GIU'}N
      { 2 SPAZI}M"
47230 B9$="{RVS}M{ 2 SPAZI}N{ 4 SIN}
      {GIU'} >{OFF}[<B>]{RVS} { 4 SIN}
      {GIU'} {OFF}[<B>]{RVS}↑ { 4 SIN}
      {GIU'}N{ 2 SPAZI}M"
47240 D3$="M{ 2 SPAZI}N{ 4 SIN}{GIU'}
      [<B>]V { 4 SIN}{GIU'} >[<B>]
      { 4 SIN}{GIU'}N{ 2 SPAZI}M"
47260 D5$="M{ 2 SPAZI}N{ 4 SIN}{GIU'}
      [<C>]V[<V>]{ 4 SIN}{GIU'}[<D>]<
      [<F>]{ 4 SIN}{GIU'}N{ 2 SPAZI}M"
47300 SB$="{ 4 SPAZI}{ 4 SIN}{GIU'}
      { 4 SPAZI}{ 4 SIN}{GIU'}{ 4 SPAZI}
      { 4 SIN}{GIU'}{ 4 SPAZI}{ 3 SU}"
47350 SP$="{ 4 DES}{ 4 SIN}{GIU'}
      { 4 DES}{ 4 SIN}{GIU'}{ 4 DES}
      { 4 SIN}{GIU'}{ 4 DES}{ 3 SU}"
47360 ST$="{ 5 SPAZI}{ 5 SIN}{GIU'}
      { 5 SPAZI}{ 5 SIN}{GIU'}{ 5 SPAZI}
      { 5 SIN}{GIU'}{ 5 SPAZI}{ 5 SIN}
      {GIU'}{ 5 SPAZI}{ 5 SIN}{ 4 SU}"
47400 RQ$="M{ 2 SPAZI}N{ 4 SIN}{GIU'} OP
      { 4 SIN}{GIU'} L@ { 4 SIN}{GIU'}N
      { 2 SPAZI}M"
47420 RV$="[< 4 B>]{ 4 SIN}{GIU'}
      [< 4 B>]{ 4 SIN}{GIU'}[< 4 B>]
      { 4 SIN}{GIU'}[< 4 B>]"
47440 RS$="M{ 2 SPAZI}N{ 4 SIN}{GIU'}
      {RVS}{ 2 SPAZI}{OFF} { 4 SIN}{GIU'}
      {RVS}{ 2 SPAZI}{OFF} { 4 SIN}
      {GIU'}N{ 2 SPAZI}M"
47460 R2$="{RVS}M{ 2 SPAZI}N{ 4 SIN}
      {GIU'} [< 2 +>] { 4 SIN}{GIU'}
      [< 2 +>] { 4 SIN}{GIU'}N{ 2 SPAZI}
      M{OFF}"
47480 RN$="{RVS}O[< 2 T>]P{ 4 SIN}{GIU'}
      [<G>]{ 2 SPAZI}[<M>]{ 4 SIN}{GIU'}
      [<G>]{ 2 SPAZI}[<M>]{ 4 SIN}{GIU'}
      L[< 2 @>]@{OFF}"
47500 X$="UI[<@>]UI{ 5 SIN}{GIU'}N*B*M
      { 5 SIN}{GIU'}M {RVS} {OFF} N
```

Gollum per C 64

nelle variabili che permettono il disegno vero e proprio delle tracce (vedi 47000).

41300 - Se la matrice AV ha un 1 nella casella in esame, allora la traccia deve essere disegnata confusa, prescindendo da DR.

41350-41800 - A seconda del valore di DR vengono fissati i valori delle variabili grafiche TN$ e TM$, che rappresentano rispettivamente la galleria aperta e la galleria aperta reverse. Quindi il sottoprogramma termina.

43000 - Questo sottoprogramma introduce e visualizza lo spostamento del giocatore dopo la scelta del punto di partenza, e cioè quando vengono usati i tasti numerici.

43020-43060 - Inizializzazione di variabili. Di quelle numeriche il significato sarà chiarito al momento dell'utilizzo. Per quanto riguarda quelle grafiche la GS$ fa disegnare una piccola circonferenza là dove il giocatore transita: CH$ la elimina, sovrapponendovi un quadrato reverse.

43120-43320 - Vengono introdotti i caratteri dall'1 al 9, con l'esclusione del 5, e il Return. La variabile CM conta i caratteri introdotti e impedisce che il Return venga battuto come primo carattere. Se viene battuto il Return come secondo carattere, significa che la mossa è a spostamento singolo, e in questo caso viene incrementato il contatore delle mosse singole VS e la variabile booleana SG viene posta a 1 (vedi 55000). Se il secondo carattere non è Return, significa che la mossa è a spostamento doppio: viene incrementata la variabile che conta le mosse doppie VD e viene posta a 1 la variabile booleana DP (vedi 55000).

43350 - Viene chiamato il sottoprogramma che, a seconda della variabile PR$ introdotta, calcola il nuovo valore di colonna e riga (vedi 52000).

43360 - Se è stato commesso un errore (ME = 1; vedi 52000), l'ingresso della PR$ deve essere ripetuto (si ha errore, ad esempio, quando si batte lo spostamento 4, trovandosi in 1,3).

43400-43500 - Si posiziona il cursore (vedi 51000), si incrementa la già vista variabile CM e si disegna il cerchio GS$.

43510 - Se il carattere introdotto è il primo (CM = 1), deve essere salvata la posizione dello spostamento parziale di una casella nel caso di mossa a spostamento doppio. Infatti, dopo la stampa del secondo cerchio sulla galleria definitiva il precedente cerchio sulla galleria di transito deve essere cancellato da CH$. La variabile VM serve per il calcolo del tipo di mossa realizzato (vedi istruzioni 43860-43900).

43550 - Se CM è inferiore a 2, viene introdotto il secondo carattere o il Return.

43700-43850 - Viene cancellato il cerchio disegnato sulla casella su cui si è passati, ma non ci si è fermati. Nel cancellare il cerchio bisogna naturalmente ridisegnare quello su cui, a sua volta, si era sovrapposto il cerchio stesso. Bisogna quindi distinguere i casi in cui la galleria era chiusa o aperta, e in questo caso se era vuota senza tracce o con tracce. Innanzitutto viene posizionato il cursore con il consueto salvataggio per CL e RW (vedi 51000). Alla istruzione 43800 si verifica che la galleria di transito sia chiusa, nel qual caso si disegna il quadrato CH$, che cancella GS$. Alla istruzione 43820 si verifica che la galleria sia aperta e con tracce, nel qual caso si decodifica la traccia (vedi 41000) e la si disegna. Alla istruzione 43830 si verifica che la galleria sia aperta, senza tracce, ma a fondo roccioso. Alla istr. 43840 si verifica che la galleria sia aperta, ma a fondo terroso. Quindi vengono ripristinati i valori di CL e RW.

43855 - Se l'ultimo carattere introdotto era Return, la mossa è semplice e il sottoprogramma termina.

43860-44000 - Se invece l'ultimo carattere è un numero, allora la mossa è doppia e quindi si passa a valutare il tipo di mossa doppia. Questa informazione servirà a Gollum per capire con quale criterio si muove il giocatore. Le variabili che contengono i due spostamenti singoli a comporre la mossa doppia sono VM e NQ.
Se VM e NQ sono diverse, significa che la mossa è del tipo retta-diagonale (o diagonale-retta), nel qual caso si incrementa il relativo contatore DF. Se VM e NQ sono uguali e una di esse ha valore 1, 3, 7 oppure 9, allora la mossa è del tipo diagonale-diagonale e si incrementa DN. Se invece non si tratta del caso precedente, allora, per esclusione, si incrementa il contatore DI, che corrisponde allo spostamento retta-retta. Quindi il sottoprogramma termina.

45000 - Questo sottoprogramma "nasconde" Gollum all'inizio della partita e determina il numero e la dislocazione delle gallerie a fondo roccioso.

45100-45220 - Vengono estratte una riga e una colonna a caso; la corrispondente posizione in SC viene posta uguale a 5 e nella quinta posizione dei due vettori OC e OW (vedi 35000) vengono posti i valori di CL e RW. I valori estratti vengono trasferiti poi in NC e NR, perché queste sono le variabili impiegate dai sottoprogrammi che utilizza Gollum.

45300-46000 - Viene fissato il numero totale TD di riquadri rocciosi, che è casuale e varia tra 5 e 15. Vengono quindi estratte TD coppie di numeri, per colonne e righe, casuali e le posizioni relative nella matrice SS vengono poste a 1. Quindi il sottoprogramma termina.

47000-47600 - In questo sottoprogramma vengono inizializzate le variabili grafiche (anziché variabili sarebbe più corretto dire, in questo caso, costanti) del programma.

48000 - Questo sottoprogramma viene attivato quando il giocatore trova Gollum.

48040-48110 - Vengono aggiunti messaggi e cancellate righe diverse, a seconda che ci si trovi in prima mossa (JJ = 0) o in mosse successive (JJ = 1).

48300-48650 - Vengono cancellate graficamente tutte le gallerie, ad eccezione di quella dove è nascosto

Gollum per C 64

Gollum, e sulla quale si trova il giocatore.

48600-50000 - Viene fatto lampeggiare il volto stilizzato di Gollum (X$) nella posizione in cui è stato trovato. Quindi il sottoprogramma termina.

50500-50950 - Questo sottoprogramma fa stampare un cursore intermittente per la Get, nel caso di domande rivolte all'utente (esempio: "Vuoi le istruzioni?").

51000 - Questo sottoprogramma, di impiego assai diffuso, posiziona il cursore sullo schermo, partendo dai valori di colonna (CL) e riga (RW).

51100-51600 - Il sottoprogramma consiste di due cicli For, che ad ogni "passata" e per CL ed RW volte spostano il cursore, in orizzontale e verticale, di una lunghezza pari al lato di una singola casella.

52000 - Questo sottoprogramma traduce la variabile PR$, vista in 43000, in modifiche sulla riga e sulla colonna della posizione del giocatore, in modo da determinare la nuova posizione.

52090 - La variabile ME è booleana, e vale 0, se la mossa dell'utente è corretta, 1, se è sbagliata. Qui viene semplicemente inizializzata.

52100-53000 - La conversione avviene in due modi leggermente differenti, a seconda che si tratti di spostamenti che prevedono la modifica della riga o della colonna o di entrambe. Nel secondo caso, infatti, la prevenzione dell'errore viene fatta sul test di conversione stesso. Facciamo un esempio: supponiamo che il giocatore si trovi nella posizione 3,4 (o C-4). Il secondo 2 lo porterebbe, per così dire, fuori dalla mappa delle 45 caselle. Dato che però il valore della colonna è rimasto costante, è sufficiente riportare la riga al valore limite 5 e segnalare l'errore, impedendo la conversione del secondo 2.

Supponiamo ora, invece, che sempre dalla 3,4 (colonna, riga) si batta prima il 2, ma poi il 3. A questo punto, per quanto riguarda il 3 solo la riga ha violato i limiti, ma non la colonna, che quindi viene ad assumere un nuovo valore. Pertanto, nella correzione il giocatore viene a trovarsi su una casella (in questo caso la 4,5) che non è più quella sulla quale si era fermato.

55000 - Questo sottoprogramma determina il contenuto di una galleria in seguito alla sua apertura.

55010 - La variabile CF è booleana e quando vale 1 significa che il gioca-

Seguito listato Gollum.

```
      { 5 SIN}{GIU'}¯M*N { 5 SIN}{ḠIU'}
      { 2 SPAZI}[<T>]{ 2 SPAZI}"
47550 BN$="{ 5 SPAZI}{ 5 SIN}{GIU'}
      { 5 SPAZI}{ 5 SIN}{GIU'}{ 5 SPAZI}
      { 5 SIN}{GIU'}{ 5 SPAZI}{ 5 SIN}
      {GIU'}{ 5 SPAZI}"
47600 RETURN
48000 REM GOLLUM TROVATO
48040 PRINT"{HOME}{ 23 GIU'}{ 32 DES}"MS+
      1
48050 TV=1
48100 IF JJ=1 THEN PRINT"{HOME}
      { 23 GIU'}{OFF}{ 14 SPAZI}"
48110 IF JJ=0 THEN PRINT"{HOME}
      { 23 GIU'}{OFF}{ 18 SPAZI}"
48300 PRINT"{HOME}{ 2 GIU'}{ 4 DES}";
48350 FOR Q=1 TO 5
48400 FOR Y=1 TO 9
48450 IF CL=Y AND RW=Q THEN PRINTSP$;:GOT
      O 48550
48500 PRINTSB$;
48550 NEXT Y
48600 PRINT"{ 3 DES}{ 3 GIU'}{DES}";
48650 NEXT Q
48700 GOSUB 51000
49120 PRINT"{SIN}";:AR=500:GOSUB 40000
49150 FOR J=1 TO 10
49200 PRINTX$;:PRINT"{ 4 SU}{ 5 SIN}";
49250 AR=150:GOSUB 40000
49300 PRINTST$;
49350 AR=150:GOSUB 40000
49400 NEXT
50000 RETURN
50500 REM CURSORE INTERM. PER LA GET
50600 PRINT"{RVS} SI {OFF}{ 4 SPAZI}
      { 8 SIN}";
50620 GET US$:IF US$="S" OR US$="N" THEN
      50950
50650 AR=300:GOSUB 40000
50700 PRINT"{OFF}{ 4 SPAZI}{RVS} NO {OFF}
      { 8 SIN}";
50720 GET US$:IF US$="S" OR US$="N" THEN
      50950
50750 AR=300:GOSUB 40000
50900 GOTO 50600
50950 RETURN
51000 REM POSIZIONAMENTO DISEGNO RIQUADRI

51100 PRINT"{HOME}{ 2 GIU'}{ 4 DES}";
51150 IF CL=1 THEN 51350
51200 FOR J=1 TO CL-1
51250 PRINT"{ 4 DES}";
51300 NEXT
51350 IF RW=1 THEN 51600
51400 FOR J=1 TO RW-1
51450 PRINT"{ 4 GIU'}";
51500 NEXT
51600 RETURN
52000 REM CONVERSIONE SPOSTAMENTI
52090 ME=0
52100 IF PR$<>"1" THEN 52150
52120 IF CL>1 AND RW<5 THEN CL=CL-1:RW=RW
      +1:GOTO 52150
52130 ME=1
52150 IF PR$="2" THEN RW=RW+1
52200 IF PR$<>"3" THEN 52250
52220 IF CL<9 AND RW<5 THEN CL=CL+1:RW=RW
      +1:GOTO 52250
52230 ME=1
52250 IF PR$="4" THEN CL=CL-1
52300 IF PR$="6" THEN CL=CL+1
52350 IF PR$<>"7" THEN 52400
52370 IF CL>1 AND RW>1 THEN CL=CL-1:RW=RW
      -1:GOTO 52400
```



Gollum per C 64

Seguito listato Gollum.

```
52380 ME=1
52400 IF PR$="8" THEN RW=RW-1
52450 IF PR$<>"9" THEN 52500
52470 IF CL<9 AND RW>1 THEN CL=CL+1:RW=RW
      -1:GOTO 52500
52480 ME=1
52500 IF CL<1 THEN CL=1:ME=1
52540 IF CL>9 THEN CL=9:ME=1
52560 IF RW<1 THEN RW=1:ME=1
52580 IF RW>5 THEN RW=5:ME=1
53000 RETURN
55000 REM GESTIONE RIQUADRI NUOVI
55010 CF=1:VT=T2
55020 IF SC(CL,RW)=5 THEN GOSUB 48000:GOT
      O 60000
55040 IF DR(CL,RW)>0 AND SC(CL,RW)>0 THEN
       GOSUB 41000:PRINTTN$:GOTO 55200
55060 IF SS(CL,RW)<>0 THEN 55080
55070 PRINTRQ$:TM$="{RVS}"+RQ$
55072 IF T1=1 AND CF=1 THEN T2=0
55075 GOTO 55300
55080 IF SS(CL,RW)=1 THEN PRINTRS$:TM$=R2
      $:T2=0:GOTO 55300
55200 T1=1:T2=1
55300 IF T1=1 AND T2=0 THEN FP=FP+1:GOTO
      55350
55320 FP=0
55350 IF FP>=5 THEN T1=0:T2=0:FP=0
56000 IF VT=1 AND DP=1 THEN Z2=Z2+1
56020 IF VT=1 AND SG=1 THEN Z1=Z1+1
56100 RETURN
56500 REM RALLENTAMENTO DI AR
56510 GET AC$:IF AC$<>"+" AND AC$<>"-" TH
      EN 56600
56520 IF AC$="-" THEN AR=AR+500:GOTO 5654
      0
56530 IF AC$="+" THEN AR=AR-500
56540 IF AR<=0 THEN AR=0
56550 GOTO 56510
56600 RETURN
57000 REM APERTURA RIQUADRI
57260 IF SA(CL,RW)=1 THEN 58500
57300 GOSUB 51000
57350 NA=NA+1
57800 FOR J=1 TO 3
57820 PRINTRV$;:PRINT"{ 3 SU}{ 4 SIN}";
57840 AR=150:GOSUB 40000
57850 PRINTRN$;:PRINT"{ 3 SU}{ 4 SIN}";
57870 AR=150:GOSUB 40000
57900 NEXT
58500 RETURN
60000 REM OPZIONI DI FINE PARTITA
60002 IF EX=1 THEN 60100
60005 PRINT"{HOME}{ 23 GIU'}{ 13 DES}
      { 27 SPAZI}";
60010 FOR T=21 TO 1 STEP-1
60015 POKE RB+(40*T),32:AR=1:GOSUB 40000
60020 NEXT
60025 FOR T=0 TO 39
60030 POKE RB+T,32:AR=1:GOSUB 40000
60035 NEXT
60100 PRINT"{CLR}";
60200 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
      :PRINT
60250 IF MS=0 THEN 60350
```

Principale 1-7000	63900		
	40000		
	47000		
	33000		
	50500	40000	
	31000		
	63000		
	51000		
	62000		
	45000		
	43000	52000	
		51000	
		41000	
	57000	51000	
		40000	
	55000	48000	51000
			40000
		41000	
	20000	29000	
		35000	
	30000	40000	
	40000		
Principale 60000-61320	15000	51000	
		40000	
		56500	

Figura 1. *Quadro generale della disposizione dei sottoprogrammi e dei livelli di chiamata.*

tore ha perduto le tracce di Gollum. La variabile VT assume il valore di T2 (che indica mossa per mossa se il giocatore è sulle tracce di Gollum) della mossa precedente.

55020 - Se il riquadro scelto, o su cui è arrivato il giocatore in seguito ad una mossa, ha nella matrice SC il valore 5, significa che Gollum è lì e che quindi è stato scoperto. Si chiama pertanto il sottoprogramma 48000, che informa l'utente del "lie-

Gli ultimi saranno i primi?

Giudicalo tu.

Anche se, in ordine di tempo, siamo gli ultimi a uscire con libri "dedicati" ai più diffusi home e personal computer, siamo convinti che il lettore attento ed esigente apprezzerà la qualità dei contenuti e la loro presentazione. Abbiamo selezionato e tradotto i best-seller mondiali che accolgono quanto di più utile è stato scritto per il tuo computer.

Scegli a colpo sicuro!

Puoi ordinare direttamente i titoli presentati compilando ed inviando il coupon pubblicato oppure acquistarli presso i più qualificati computer shop e le migliori librerie.

CEDOLA DI ORDINAZIONE - LIBRI

Compilare e spedire in busta chiusa a:
J. soft - Viale Restelli 5 - 20124 Milano
tel. 02/6880841-6880842-6880843

Ordino i seguenti libri per un importo totale di L. ....
+ L. 2.000 come contributo fisso per spese di spedizione

Cod. .... Cod. ....
Cod. .... Cod. ....
Cod. .... Cod. ....

☐ Contanti allegati
☐ Assegno allegato n° ....
☐ Ho spedito l'importo a mezzo vaglia postale
☐ Ho versato l'importo sul CCP n° 19445204 intestato a J. soft - Milano
☐ Pagherò in contrassegno al postino al ricevimento dei volumi

Nome ....
Cognome ....
Via ....
CAP .... Città .... Prov. ....
Se richiesta fattura - codice fiscale ....
Data ....

J. soft: libri di qualità

Per Commodore 64

Giochi fantastici per il C 64
Cod. ASOC 001 L. 42.000
(inclusa cassetta)

Un libro pensato e realizzato per il divertimento dell'utente del C 64; contiene 19 giochi di vario genere tutti ampiamente commentati. L'attento studio dei programmi potrà essere vantaggioso per l'apprendimento delle tecniche di programmazione.

Il libro del C 64
Volume 1
Cod. ASOC 010 L. 24.000
(inclusa cassetta)
Il libro del C 64
Volume 2
Cod. ASOC 011 L. 24.000
(inclusa cassetta)

I due libri illustrano a fondo le possibilità del Commodore 64. Un compendio di utili consigli, e quant'altro è necessario per conoscere meglio il proprio calcolatore. Esempi pratici completano l'esposizione in modo chiaro ed esauriente.

C 64: suono e grafica -
Volume 1
Cod. ASOC 008 L. 24.000
(inclusa cassetta)
C 64: suono e grafica -
Volume 2
Cod. ASOC 009 L. 24.000
(inclusa cassetta)

Due volumi che non possono mancare nella biblioteca dell'utente Commodore. Una miniera di idee e suggerimenti per la programmazione del tuo personal computer. Una vasta serie di programmi esemplificativi guidano il lettore al miglior sfruttamento del Commodore 64.



Per ZX Spectrum

15 giochi grafici per ZX Spectrum
Cod. ASOC 002 L. 24.000
(inclusa cassetta)

Un fantastico volume con 15 entusiasmanti giochi grafici per il tuo ZX Spectrum. Tutti i programmi sono "pronti all'uso" e ampiamente commentati. I programmi sono scritti sfruttando appieno le capacità del calcolatore ed il loro studio può giovare a chiunque intenda affinare le proprie tecniche di programmazione.

Inoltre, i Paper Book: la raccolta dei programmi suddivisi per macchina di Paper Soft, il primo settimanale di software su carta per il tuo computer.



Paper Book - programmi per Apple // -
Cod. ASOC 005 L. 18.000
Paper Book - programmi per VIC 20 -
Cod. ASOC 003 L. 18.000
Paper Book - programmi per Commodore 64 -
Cod. ASOC 004 L. 18.000
Paper Book - programmi per Texas TI 99/4A -
Cod. ASOC 007 L. 18.000
Paper Book - programmi per ZX Spectrum -
Cod. ASOC 006 L. 18.000

In ogni volume, corredato di cassetta, oltre 40 programmi per il tuo computer: giochi grafici, utility, arcade, adventure una miscela esplosiva di software "pronto all'uso"!

J. soft s.r.l.
Viale Restelli, 5 - 20124 Milano - tel. 02/6880841-6880842-6880843

PERSONAL SOFTWARE

Gollum per C 64

Seguito listato Gollum.

```
60300 PRINT"{ 8 DES}VUOI RIVEDERE IL PERC
      ORSO ?":PRINT:PRINT:PRINT
60350 PRINT"{ 8 DES}VUOI GIOCARE ANCORA ?
      ":PRINT:PRINT:PRINT
60400 PRINT"{ 8 DES}NON VUOI GIOCARE PIU'
       ?":PRINT:PRINT:PRINT
60500 AR=400:GOSUB 40000
60600 PRINT"{HOME}{ 5 GIU'}";
60620 IF MS=0 THEN 60700
60650 PRINT"{GIU'}{ 3 DES}[< 3 I>]{GIU'}
      { 3 SIN}{RVS} 1 {GIU'}{ 3 SIN}
      [< 3 I>]{OFF}":AR=500:GOSUB 40000
60700 PRINT"{GIU'}{ 3 DES}[< 3 I>]{GIU'}
      { 3 SIN}{RVS} 2 {GIU'}{ 3 SIN}
      [< 3 I>]{OFF}":GOSUB 40000
60750 PRINT"{GIU'}{ 3 DES}[< 3 I>]{GIU'}
      { 3 SIN}{RVS} 3 {GIU'}{ 3 SIN}
      [< 3 I>]{OFF}":GOSUB 40000
61000 GET RI$:IF RI$="" THEN 61000
61100 IF RI$="1" AND MS<>0 THEN GOSUB 150
      00:GOTO 60100
61200 IF RI$="2" THEN PP=1:GOTO 20
61300 IF RI$<>"3" THEN 61000
61310 PRINT"{CLR}{ 10 GIU'}":PRINT"
      { 15 DES}A PRESTO !{ 5 GIU'}"
61320 END
62000 REM COMPLETAMENTO RIQUADRI
62010 PRINT"{HOME}{ 2 GIU'}";
62020 FOR T=1 TO 5
62040 PRINT"{ 4 DES}";
62060 FOR C=1 TO 9
62080 PRINT"{RVS}O[< 2 T>]P{ 4 SIN}{GIU'}
      ";
62100 PRINT"{RVS}[<G>]{ 2 SPAZI}[<M>]
      { 4 SIN}{GIU'}";
62120 PRINT"{RVS}[<G>]{ 2 SPAZI}[<M>]
      { 4 SIN}{GIU'}";
62140 PRINT"{RVS}L[< 2 @>]@{ 3 SU}";
62160 NEXT
62180 PRINT"{ 3 GIU'}";
62200 NEXT
62220 RETURN
63000 REM PRESENTAZIONE INIZIALE
63060 QQ=0
63080 FOR C=1 TO 10
63100 RW=INT(RND(1)*14)+4
63200 CL=INT(RND(1)*30)+6
63300 FOR Z=1 TO RP:PRINT"{GIU'}";:NEXT
63350 FOR Z=1 TO CP:PRINT"{DES}";:NEXT
63400 IF QQ=1 THEN PRINTBN$
63480 PRINT"{HOME}";
63500 FOR Z=1 TO RW:PRINT"{GIU'}";:NEXT
63600 FOR Z=1 TO CL:PRINT"{DES}";:NEXT
63700 PRINTX$;
63720 RP=RW
63740 CP=CL
63750 QQ=1
63770 PRINT"{HOME}";
63800 NEXT
63850 RETURN
63900 REM RICONOSCIMENTO 4032/C-64
63910 IF PEEK(40961)=160 THEN RB=32768
63920 IF PEEK(40961)=227 THEN RB=1024:POK
      E 53280,6:POKE 53281,6
63930 IF RB=0 THEN STOP
63940 RETURN
```

to evento" e quindi si passa alle opzioni finali (vedi 60000).

55040 - Si disegna la traccia corrispondente nel caso in cui SC abbia un valore compreso tra 1 e 4. Il tipo di traccia è fissato, al solito, da DR (vedi 41000).

55060-55080 - A seconda del valore di SS(CL,RW), si disegna una galleria a fondo roccioso o terroso. In entrambi i casi la variabile T2 va, o resta, a 0. La differenza fra T1 e T2 è la seguente: T2 precisa se l'utente è sulle tracce di Gollum mossa per mossa; T1 è anch'essa booleana, ma non va a 0 non appena il giocatore perde le tracce, ma dopo 5 mosse, se nel frattempo, ovviamente, non le ritrova. Infatti, per un certo numero di mosse, anche se ha smarrito le tracce, il giocatore è ancora in grado di intuire gli spostamenti di Gollum e quindi di essere pericoloso.

55200 - Si arriva a questa istruzione quando le tracce sono state trovate.

55300-55350 - Se il giocatore ha perso le tracce di Gollum, parte il contatore FP, che giunto al 5 farà azzerare, come si è già detto, anche il valore di T1.

56000-56100 - Se il giocatore era sulle tracce prima di realizzare la mossa (VT = 1), e la mossa è doppia, allora viene incrementato il contatore Z2, altrimenti, se è singola, viene incrementato Z1 (vedi 43000; vedi 20000). Quindi il sottoprogramma termina.

56500-56600 - Questo sottoprogramma permette di modificare, durante l'esecuzione del riepilogo del percorso, il valore dei tempi di pausa e quindi di modificare il valore della variabile AR (vedi 40000) che li determina. Il procedimento è semplice: se viene battuto "+", la variabile AR si decrementa (rallenta il tempo di pausa e quindi cresce la velocità); se viene battuto "–", si incrementa.

57000 - Questo sottoprogramma mostra l'apertura di una galleria.

57260-58500 - Se la galleria è già aperta, il sottoprogramma termina. Altrimenti, dopo aver incrementato la variabile NA (che conta il numero totale delle gallerie aperte), viene fatto lampeggiare per 3 volte un riquadro a scacchiera che precede l'apertura, in modo da dare un po' di "suspance".

60000 - Qui riprende il programma principale nella sua parte conclusiva.

60002 - Se Gollum non è stato trovato (EX = 1), si salta alla 60100.

60005-60035 - Viene cancellato il bordo dello schermo.

60100-61320 - Viene presentato il menu delle diverse possibilità che ha il giocatore al termine della partita. Se Gollum è stato trovato alla stessa scelta del punto di partenza (caso raro, ma non impossibile), non viene data la possibilità 1 di rivedere il percorso (MS = 0). Se il giocatore



Gollum per C 64

vuole rivedere il percorso (in tutti gli altri casi), viene chiamato il sottoprogramma 15000 e quindi si torna alla istruzione 60100. Se vuole giocare ancora, si pone a 1 la variabile booleana PP, che evita la presentazione iniziale. Se non vuole più giocare, il programma termina.

62000-62220 - Questo sottoprogramma completa le gallerie nel disegno random iniziale delle stesse. Questo viene fatto per non allungare troppo i tempi della preparazione dello schermo per il gioco. Vengono stampate tutte le gallerie, che vanno così a riempire i "buchi" eventualmente lasciati dalla preparazione casuale (vedi istruzioni 1100-1250).

63000 - Questo sottoprogramma viene impiegato nella presentazione, dopo il titolo e prima del disegno delle gallerie. Esso fa semplicemente disegnare per 10 volte il volto di Gollum (X$), che appare e scompare in diversi punti dello schermo.

63080-63850 - Essendo la variabile X$, che rappresenta il volto di Gollum, leggermente più larga dei riquadri per motivi puramente estetici, in questo sottoprogramma non è stato possibile impiegare il sottoprogramma 51000 di posizionamento del cursore, ed è stato realizzato un apposito ciclo interno ad uso locale. Viene disegnata la X$, poco dopo cancellata dalla BN$, quindi ancora la X$ in un'altra posizione e così via.

63900 - Questo sottoprogramma determina il registro base dello schermo per tutte quelle istruzioni che usano la Poke grafica, che, come si sa, varia dal Commodore al C 64. Il procedimento è assai semplice: vengono lette due locazioni note della ROM e in base a queste si determina RB. Per il C 64 vengono fissati anche il colore blu per il bordo e lo sfondo che rende chiaro lo svolgimento del gioco. Il programma, nato e sviluppato su Commodore, è purtroppo monocromatico, ma gli utenti del C 64 crediamo non avranno grosse difficoltà a dargli un po' di colore.

■

... e con questi siamo a 80 ecco chi si è abbonato, ha risparmiato e ha vinto un Commodore 64

Pubblichiamo l'elenco di altri 20 fortunati vincitori del concorso abbonamenti Jackson 1985. A loro vanno le nostre più vive felicitazioni e agli altri nostri abbonati l'augurio di essere fra i 20 restanti fortunati.
Il prossimo mese pubblicheremo l'elenco degli ultimi 20 fortunati vincitori.



1) **LUNGHI Giuseppe**
Via Gramsci, 5
26013 Crema (CR)

2) **I.P.S.I.A. "Giorgi"**
Via Tenaglio, 53
31100 Treviso (TV)

3) **LIGNOCCHI Roberto**
Via B. Bonini, 11
25127 Brescia (BS)

4) **BRIZZI Andrea**
V.le Europa, 18
39100 Bolzano (BZ)

5) **MICRO CONTROL snc**
Via Claudia, 4557
41056 Savignano Sul Panaro (MO)

6) **SPARACO Lucio**
Via Aldo Moro, 169
81055 Santa Maria Capua Vetere (CE)

7) **OBERTINO Silvio**
Via Valle Soana, 59
10085 Pont Canavese (TO)

8) **CHIAVIERI Renato**
P.zza Mazzini, 20
44030 Serravalle (FE)

9) **ALESSANDRELLI Fabio**
Via Fano, 35
60128 Ancona (AN)

10) **CASSAI Fulvio**
Via Papa Giovanni XXIII, 36
20097 Melegnano (MI)

11) **BIANCHERI Giovanni**
Via S. Rufino, 16
16040 Leivi (GE)

12) **MANGIARACINA Antonio**
Via Vill. Messina, 5
92018 S. Margherita di Belice (AG)

13) **SCOTTA Eligio**
Via Castello, 65
12039 Verzuolo (CN)

14) **CACUDI Alessandro**
Via Benanduci, 6
72021 Francavilla Fontana (BR)

15) **BURZAGLI Laura**
Via Traversi, 7
50127 Firenze (FI)

16) **DI STEFANO Gaetano**
Via Scarcella, 67
95030 Trappeto (CT)

17) **PIZZIRANI Luigi**
Via Pessina, 34
70125 Bari (BA)

18) **PIANA Guido**
Via Petrarca, 39
07021 Arzachena (SS)

19) **OLIVARI Claudio**
Via Piave, 6
46034 Governolo (MN)

20) **SCHIESARO Roberto**
Via Porta PO, 47
45100 Rovigo (RO)





Skywalker per C 64

Un gioco che sicuramente vi appassionerà

di *Luca Marras*

Siamo nel Medioevo. Una strega malvagia ha fatto un incantesimo: presto una minacciosa nube nera coprirà la Luna simbolo del Bene. Ma come ogni sortilegio che si rispetti, anche questo può essere sciolto da un uomo coraggioso. E questa volta l'ingrato compito tocca proprio a voi. L'oscuramento della luna sarà preceduto da una pioggia di stelle. Voi, sul vostro cavallo volante, dovrete acchiapparle tutte. Non lasciatevene sfuggire neppure una! Infatti ogni volta che una stella cadrà a terra, la nube si avvicinerà sempre più al disco lunare....

Il vostro viaggio è tutt'altro che facile: infatti vi vengono incontro dei coloratissimi uccelli inviati dalla fattucchiera, che vi uccideranno al solo contatto (per fortuna avete ben tre vite a disposizione!). Inoltre dovrete stare molto attenti a non toccare il suolo, e soprattutto state il più possibile lontani dalla freccia che la strega vi scaglia contro dal profondo della terra.

Nel cielo si trova anche un misterioso scrigno che vi regalerà un mucchio di punti. Dalla vostra avrete anche un contro-incantesimo, che vi renderà invulnerabili ai rapaci. Per azionarlo, basta premere "Fire" sul joystick. Questo incantesimo non è però di durata illimitata: esso è disponibile finchè gli angoli superiori dello schermo sono verdi. Se diventano gialli, l'incantesimo sta per finire. Se sono rossi esso è già scaduto. Alla fine di ogni livello dispari si ha un bonus variabile e una vita in regalo (ma la dotazione non potrà mai essere superiore alle tre vite).

Listato 1. *Il programma Skywalker.*

```
1 REM "SKYWALKER" BY LUCA MARRAS
9 POKE808,225
10 POKE53280,0:POKE53281,0:DIMN,HO,WA$(10
   ,2),SC$(24):V=53248:POKEV+21,0
15 PRINTCHR$(8)CHR$(142)
20 SI=54272:VL=SI+24:POKEVL,0:FORP=1TO3:U
   =SI+7*(P-1):PRINT"{CLR}"
30 FL(P)=U:FH(P)=U+1:TL(P)=U+2:TH(P)=U+3:
   W(P)=U+4:AD(P)=U+5:SR(P)=U+6:NEXT
40 FORP=1TO3:POKEW(P),0:NEXT:POKEVL,15
50 POKEAD(1),4*16+8:POKESR(1),16+10:HI$="
   L.M.S."
60 POKEAD(2),6*16:POKESR(2),10*16+10:POKE
   TL(2),200:POKETH(2),4:CO$="{ 2 YEL}
   {RED}"
70 POKEAD(3),10*16+10:POKESR(3),0:LI$(0)=
   "{ 2 SPAZI}":LI$(1)="{PUR}← ":LI$(2)="
   {PUR}←{RED}←"
100 GOSUB1000:HI=1985
110 FORP=2TO10:WA$(P,1)=" ":WA$(P,2)=" ":
    NEXT:WA$(1,1)="<":WA$(1,2)=">"
200 POKE53270,PEEK(53270)AND239:POKE53272
    ,21:POKE198,0
210 PRINT"{CLR}{ 4 GIU'}{GRN} U{ 2 C} -
    - - - -{ 3 SPAZI}- UCI -{ 3 SPAZI}-
    - U{ 2 C} [<A>]CI"
220 PRINT" -{ 3 SPAZI}- - - - -
    { 3 SPAZI}- - - -{ 3 SPAZI}- - -
    { 3 SPAZI}- -"
230 PRINT" JCI [<Q>]CK J[<R>]K - - -
    [<Q>]C[<W>] -{ 3 SPAZI}[<Q>]CK [<Q>]
    C{ 2 SPAZI}[<Q>]CK"
240 PRINT"{ 3 SPAZI}- - B{ 2 SPAZI}B
    { 2 SPAZI}B B B B B B{ 3 SPAZI}B B B
    { 3 SPAZI}B B"
250 PRINT" { 2 C}K -{SPAZI}-{ 2 SPAZI}-
    { 2 SPAZI}JC[<E>]CK - - J{ 2 C} - -
    J{ 2 C} - -"
260 PRINTSPC(10)"{ 4 GIU'}{RED}WRITTEN BY
     L. MARRAS"
265 PRINTSPC(11)"{PUR}{GIU'}JOYSTICK IN P
    ORT 2"
270 PRINTSPC(10)"{GIU'}{WHT}F1{BLU} OR
    {WHT}FIRE{BLU}{ 2 SPAZI}TO START"
275 PRINT"{HOME}[<6>]"TAB(5)HI$;TAB(29)HI
    ;
280 GETA$:SYS828:IFPEEK(815)<>0ANDA$<>"
    {F1}"THEN280
290 PRINT"{CLR}"
300 SG=1:Q=0:I=2:HO=201
390 POKE53272,(PEEK(53272)AND240)+14:POKE
    53270,PEEK(53270)OR16
400 POKEV,32:POKEV+1,162:GOSUB2000:POKEV+
    4,32:POKEV+5,254:POKEV+10,240
410 POKEV+11,60:POKEV+8,30:POKEV+9,60:POK
    EV+21,127:CO=0:MT=0
450 PRINT"{CLR}":FORP=1TO24:PRINTSC$(P);:
    IFP=20THENPRINTMID$(CO$,SGAND3,1);
455 NEXT:POKE56295,7:POKE49799,0:IF(SGAND
    3)=3THENPOKE56295,2
460 PRINT"{HOME}{DES}{GRN}SCORE"Q;TAB(14)
    LI$(I)TAB(30)"{CYN}ROUND"SG
470 PRINT"{HOME}{GRN}←"TAB(39)"←"
475 IFCO>89THENPRINT"{HOME}{YEL}←"TAB(39)
    "←"
480 IFCO>99THENPRINT"{HOME}{RED}←"TAB(39)
    "←"
490 PK=PEEK(V+30):PK=PEEK(V+31)
500 FORN=1TO2:SYS49700:SYS49152:SYS828:SY
    S900:SYS950:SYS950:SYS975
505 HO=401-HO:POKE2040,HO:IFSG>2THENSYS95
    0:SYS975
520 IFPEEK(V+3)=>240THEN7000
```



Skywalker per C 64

Seguito listato Skywalker.

```
530 PK=PEEK(V+30):IF(PKAND1)=1THEN6000
680 IFCO<101ANDPEEK(815)=0THENGOSUB2050:N
    =1:NEXT
690 IF(PEEK(V+31)AND1)=1THEN8000
700 N=1:NEXT
1000 DATA169,0,141,2,220,173,0,220
1020 DATA141,46,3,41,16,141,47,3,173,46,3
     ,41,15,141,46,3,169,15,237,46,3
1025 DATA141,46,3,169,255,141,2,220,96
1030 A=828
1040 READB:POKEA,B:IFB=96THEN1060
1050 A=A+1:GOTO1040
1060 A=49152
1070 READB:POKEA,B:IFB=96THEN1120
1080 A=A+1:GOTO1070
1090 DATA162,2,189,148,192,133,251,189,14
     9,192,133,252,134,253,162,0,161,251
1100 DATA141,255,3,166,253,160,1,177,251,
     136,145,251,200,200,192,40,240,3,76,
     25
1110 DATA192,160,39,173,255,3,145,251,232
     ,232,224,50,240,3,76,2,192,96
1120 A=49300:FORP=1024TO1984STEP40:HI=INT
     (P/256):LO=P-HI*256
1130 POKEA,LO:POKEA+1,HI:A=A+2:NEXT
1140 A=950
1150 READB:POKEA,B:IFB=96THEN1185
1160 A=A+1:GOTO1150
1170 DATA174,2,208,224,254,240,1,232,142,
     2,208,174,3,208,232,142,3,208
1180 DATA142,40,208,96
1185 A=975
1186 READB:POKEA,B:IFB=96THEN1190
1187 A=A+1:GOTO1186
1188 DATA174,5,208,202,202,202,202,224,10
     ,208,8,174,0,208,142,4,208,162,254
1189 DATA142,5,208,96
1190 A=49700
1191 READB:POKEA,B:IFB=96THEN1200
1192 A=A+1:GOTO1191
1193 DATA169,0,141,4,212,172,135,194,185,
     136,194,141,0,212,200,185,136,194
1194 DATA141,1,212,200,192,32,208,2,160,0
     ,140,135,194,169,17,141,4,212,96
1200 A=900
1210 READB:POKEA,B:IFB=96THEN1250
1220 A=A+1:GOTO1210
1230 DATA173,46,3,41,1,201,1,208,3,32,92,
     193,173,46,3,41,2,201,2,208,3
1231 DATA32,142,193,173,46,3,41,4,201,4,2
     08,3,32,192,193,173,46,3,41,8,201,8
1232 DATA208,3,32,242,193,96
1250 A=49500
1260 READB:POKEA,B:IFB=96THEN1290
1270 A=A+1:GOTO1260
1280 DATA173,1,208,233,8,201,58,176,2,169
     ,58,141,1,208,96
1290 A=49550
1300 READB:POKEA,B:IFB=96THEN1330
1310 A=A+1:GOTO1300
1320 DATA173,1,208,105,8,201,226,144,2,16
     9,226,141,1,208,96
1330 A=49600
1340 READB:POKEA,B:IFB=96THEN1370
1350 A=A+1:GOTO1340
1360 DATA173,0,208,233,8,201,24,176,2,169
     ,24,141,0,208,96
1370 A=49650
1380 READB:POKEA,B:IFB=96THEN1405
1390 A=A+1:GOTO1380
1400 DATA173,0,208,105,8,201,240,144,2,16
     9,240,141,0,208,96
1405 FORA=1000TO1019:READB:POKEA,B:NEXT
1407 DATA174,5,208,224,254,208,1,96,202,2
     02,202,202,202,202,202,202
1408 DATA142,5,208,96
1410 POKE56334,PEEK(56334)AND254:POKE1,PE
     EK(1)AND251
1420 FORP=0TO63:FORU=0TO7:POKE14336+P*8+U
     ,PEEK(53248+P*8+U):NEXTU,P
```

Nel secondo livello, lo scenario cambia. Vi trovate, con il vostro cavallo, sulla superficie azzurra del mare. Dovrete evitare, saltando (azionando il joystick verso l'alto), delle punte velenose che affiorano dall'acqua. Inoltre dovrete anche cercare di colpire con una freccia un uccelletto verde che vi regalerà preziosi punti. Premendo il tasto "Fire" sul joystick si carica la freccia: rilasciandolo la si lancia (in questo caso, dunque, l'auto-fire di cui sono dotati alcuni joystick è del tutto inutile).

Ogni volta che verrete colpiti da una punta avvelenata perderete una vita. Al termine di questo livello vi sarà regalata una vita extra (ma niente bonus!).

Il terzo round è simile al primo. La freccia però è molto più veloce, e anche la stella cade più velocemente. Da qui in poi la difficoltà si stabilizza.

Il movimento del cavaliere, ottenuto con il joystick in porta 2, è verticale, orizzontale e diagonale nel primo livello, e solo verticale (salto) nel secondo. Non è possibile utilizzare la tastiera per giocare.

Il listato

9 - Disinserisce il tasto Run Stop. Per reinserirlo sarà sufficiente cancellare questa linea.

10-70 - Inizializzano le variabili e lo schermo, preparano il circuito sonoro.

100 - Questa linea chiama le subroutine che inizializzano la grafica e caricano il linguaggio macchina.

110 - Prepara alcune variabili usate per la grafica.

200-390 - Vengono eseguite all'inizio di ogni partita. Visualizzano lo schermo di presentazione. Inizializzano alcune variabili che controllano il flusso del gioco. Predispongono la grafica in Multicolor.

400-490 - Vengono eseguite all'inizio di ogni livello dispari e visualizzano il campo di gioco.

500-700 - Sono il "cuore" dei livelli dispari. Il loro compito principale è chiamare le routine in linguaggio macchina e controllare se si verificano collisioni sprite-sprite o sprite-sfondo.

1000-1408 - Caricano in memoria le routine in linguaggio macchina.

1410-1499 - Creano i caratteri speciali in Multicolor.

1500-1740 - Caricano in SC (x) (dove x è compreso fra 1 e 24) le 24 righe dello schermo di gioco.

Skywalker per C 64

Seguito listato Skywalker.

```
1430 POKE1,PEEK(1)OR4:POKE56334,PEEK(5633
     4)OR1
1440 POKE53272,(PEEK(53272)AND240)+14:POK
     E53270,PEEK(53270)OR16
1450 POKE53282,7:POKE53283,2:READA:IFA=-1
     THEN1500
1460 FORP=0TO7:READB:POKE14336+A*8+P,B:NE
     XT:GOTO1450
1470 DATA0,0,0,0,0,0,0,15,31,42,1,5,6,22,
     26,26,42,255
1471 DATA30,160,160,128,128,0,0,0,0,58,0,
     0,0,0,0,0,0
1472 DATA59,255,63,15,2,2,0,0,0,61,252,25
     2,240,64,64,0,0,0
1473 DATA31,255,255,255,255,255,255,255,2
     55,37,0,0,0,0,3,7,15,63
1474 DATA38,3,31,63,239,223,191,127,127,3
     9,0,0,3,7,13,27,127,255
1475 DATA33,96,184,204,247,249,159,247,25
     5,35,0,0,0,0,192,224,112,44
1476 DATA36,0,192,224,176,200,230,249,255
     ,40,255,255,239,223,255,255,255,255
1477 DATA60,2,2,42,2,2,14,59,254,62,128,1
     28,168,128,128,224,184,238
1499 DATA-1
1500 DATA"[<7>]{ 3 SPAZI}@*↑{ 28 SPAZI}@*
     ↑{ 3 SPAZI}"
1510 DATA"{ 3 SPAZI}:;={ 6 SPAZI}@*↑
     { 19 SPAZI}:;={ 3 SPAZI}"
1520 DATA"{ 12 SPAZI}:;={ 25 SPAZI}"
1530 DATA"{ 37 SPAZI}@*↑"
1540 DATA"{ 22 SPAZI}@*↑{ 12 SPAZI}:;="
1550 DATA"{ 22 SPAZI}:;={ 15 SPAZI}"
1560 DATA"[<6>]{ 40 SPAZI}"
1570 DATA"{ 40 SPAZI}"
1580 DATA"{ 7 SPAZI}@*↑{ 30 SPAZI}"
1590 DATA"{ 7 SPAZI}:;={ 18 SPAZI}@*↑
     { 9 SPAZI}"
1600 DATA"{ 28 SPAZI}:;={ 9 SPAZI}"
1610 DATA"[<5>]{ 40 SPAZI}"
1620 DATA"{ 40 SPAZI}"
1630 DATA"{ 40 SPAZI}"
1640 DATA"{ 13 SPAZI}@*↑{ 24 SPAZI}"
1650 DATA"{ 13 SPAZI}:;={ 24 SPAZI}"
1660 DATA"[<4>]{ 2 SPAZI}@*↑{ 23 SPAZI}@*
     ↑{ 9 SPAZI}"
1670 DATA"{ 2 SPAZI}:;={ 23 SPAZI}:;=
     { 9 SPAZI}"
1680 DATA"{ 40 SPAZI}"
1690 DATA"{ 40 SPAZI}"
1700 DATA"{ 8 SPAZI}%&!#{ 6 SPAZI}%&!#
     { 4 SPAZI}%&!#{ 10 SPAZI}"
1710 DATA"{ 7 SPAZI}'←←←←${ 4 SPAZI}'←←←←
     ${ 2 SPAZI}'←←←←${ 9 SPAZI}"
1720 DATA"{ 2 SPAZI}%&←(←←←←←←←←$'←←←←(←←
     ←←←←←←←←←←(←←!#{ 2 SPAZI}"
1730 DATA" '←←←(←←←←←←←←←←←←←←←←←←←←←←←←←
     ←←←(←←←$"
1740 FORP=1TO24:READSC$(P):NEXT
1750 DATA0,40,0,0,36,0,0,36,0,0,8,0,0,60,
     0,0,52,32,0,52,168,0,52,170,0,61,168
1760 DATA130,190,128,34,175,128,10,171,16
     0,2,169,96,10,0,40,40,0,8,160,0,10
1770 DATA128,0,2,128,0,2,0,0,2,0,0,0,0,0,
     0
1780 FORP=0TO62:READB:POKE64*200+P,B:NEXT
     :POKE2040,200:POKEV+28,79
1790 POKEV+39,15:POKEV+37,4:POKEV+38,9:PO
     KEV+29,16
1800 DATA0,40,0,0,36,0,0,36,0,0,8,0,0,60,
     0,0,52,32,0,52,168,0,52,170,0,61,168
1810 DATA130,190,128,34,175,128,10,171,16
     0,2,169,96,10,0,40,2,0,32,2,0,32
1820 DATA2,128,160,0,128,128,0,128,128,0,
     34,0,0,34,0
1830 FORP=0TO62:READB:POKE64*201+P,B:NEXT
     :POKEV+40,1:POKE2041,202
1840 DATA0,0,0,0,0,0,0,0,160,0,0,40,0,0
     ,42,136,0,42,169,0,10,169,96,10,149,
     96
1850 DATA2,90,144,0,155,152,0,27,144,0,42
     ,144,0,42,80,0,5,80,0,0,0,0,0,0,0,0,
     0
1860 DATA0,0,0,0,0,0,0,0,0
1870 FORP=0TO62:READB:POKE64*202+P,B:NEXT
     :POKEV+27,114:POKEV+41,8:POKE2042,20
     3
1880 DATA0,32,0,0,32,0,0,168,0,0,168,0,2,
     94,0,2,94,0,10,237,128,10,237,128
1890 DATA0,32,0,0,32,0,0,32,0,0,32,0,0,32
     ,0,0,32,0,0,32,0,0,32,0,0,32,0,0,32,
     0
1900 DATA0,32,0,0,32,0,0,32,0
1910 FORP=0TO62:READB:POKE64*203+P,B:NEXT
     :POKE2043,200:POKEV+42,7
1920 DATA0,127,0,1,223,192,3,239,224,7,24
     7,240,15,255,120,31,126,124,28,60,60
1921 DATA62,24,158,63,62,254,51,159,246,5
     5,255,254,63,255,254,31,254,124
1922 DATA27,228,60,15,240,120,15,249,248,
     7,191,240,3,255,224,1,249,192
1923 DATA0,126,0,0,0,0
1924 FORP=0TO62:READB:POKE64*204+P,B:NEXT
     :POKE2045,204:POKEV+44,7
1930 DATA0,0,0,0,0,0,0,0,0,0,0,0,56,0,0
     ,253,128,99,251,240,247,247,236
1931 DATA127,199,222,63,191,223,127,127,6
     3,62,254,254,30,255,126,0,199,124
1932 DATA29,48,120,127,224,160,255,192,0,
     59,128,0,0,0,0,0,0,0,0,0
1933 FORP=0TO62:READB:POKE64*205+P,B:NEXT
     :POKE2044,205:POKEV+43,11
1940 DATA6430,8101,9634,12860,11457,10814
     ,9634,8583
1941 DATA7217,8583,10814,12860,5728,7217,
     8583,11457
1942 FORP=0TO15:READB:HI=INT(B/256):LO=B-
     HI*256
1943 POKE49800+P*2,LO:POKE49801+P*2,HI:NE
     XT
1950 DATA0,0,0,0,0,0,0,40,0,0,255,0,2,255
     ,128,15,255,240,63,255,252,247,119,1
     19
1951 DATA255,255,255,255,255,255,170,170,
     170,255,255,255,255,255,255
1952 DATA247,119,119,63,255,252,15,255,24
     0,2,255,128,0,255,0,0,40,0,0,0,0,0,0
     ,0
1953 FORP=0TO62:READB:POKE64*206+P,B:NEXT
     :POKE2046,206:POKEV+45,10:RETURN
2000 POKEV+3,15:POKEV+2,INT(RND(1)*256)
2010 POKEV+12,INT(RND(1)*231)+24:POKEV+13
     ,INT(RND(1)*188)+52:RETURN
2050 POKEV+39,2:CO=CO+1:IFCO>89THENPRINT"
     {HOME}{YEL}←"TAB(39)"←"
2060 IFCO>99THENPRINT"{HOME}{RED}←"TAB(39
     )"←"
2070 POKEV+39,1:PK=PEEK(V+31):RETURN
3000 POKEV+21,0:PRINT"{CLR}{DES}{GRN}SCOR
```



Skywalker
per C 64

Seguito listato Skywalker.

```
     E"Q;TAB(14)LI$(I)TAB(30)"{CYN}ROUND"
     SG:POKE49799,0
3010 PRINT"{ 16 GIU'}{BLU}";:FORP=1TO6
3020 PRINT"←←←←←←←←←←←←←←←←←←←←←←←←←←←←←
 –   ←←←←←←←←←←";:NEXT:POKEV+10,240
3030 POKEV+11,60:POKEV,80:POKEV+1,164:MT=
     0
3040 PRINT"{HOME}{ 16 GIU'}[<7>]
     { 40 SPAZI}";
3050 PRINT"{HOME}{ 5 GIU'}[<6>]
     { 17 SPAZI}@*↑{ 20 SPAZI}";
3060 PRINT"{ 17 SPAZI}:;={ 20 SPAZI}"
3070 POKEV+4,80:POKEV+5,254:POKEV+21,37
3090 PK=PEEK(V+30):PK=PEEK(V+31):CO=7:R=2

3100 FORN=1TO2:FORWA=1TO2:SYS49700:SYS491
     52:SYS828:IFPEEK(814)=1ANDCO>6THENCO
     =0
3105 SYS1000
3110 HO=401-HO:POKE2040,HO:CO=CO+1:IFCO<7
     THENPOKEV+1,154:IFCO=6THENPOKEV+1,16
     4
3120 MT=MT+1:IFMT=200THEN11500
3130 IFPEEK(815)=0THENPOKEV+5,158
3140 PRINT"{HOME}{ 16 GIU'}{BLU}"SPC(39)W
     A$(R,WA);
3150 PK=PEEK(V+31):IF(PKAND1)=1THEN8500
3160 IF(PKAND4)=4THENGOSUB6500
3200 NEXT:R=INT(RND(1)*10)+1:N=1:NEXT
6000 IF(PKAND2)=2THEN6050
6010 IF(PKAND64)=64THEN6150
6020 IF(PKAND4)=4THEN6100
6030 GOTO680
6050 Q=Q+150:PRINT"{HOME}{DES}{BLU}SCORE"
     Q:POKEW(2),0:POKEW(2),65
6055 FORP=50TO250:POKEFH(2),P:NEXT:POKEW(
     2),0
6060 GOSUB2000:PK=PEEK(V+30):MT=MT+1:IFMT
     =10THEN11000
6070 GOTO680
6100 IF(PKAND16)=16THEN680
6105 FORP=1TO10:POKEV+41,P:POKEW(1),0:POK
     EFH(1),150:POKEW(1),129
6110 FORU=1TO20:NEXTU,P:POKEW(1),0:POKEV+
     41,8:GOTO8000
6150 Q=Q+200:PRINT"{HOME}{DES}{BLU}SCORE"
     Q:POKEV+45,1:FORP=1TO10:POKEW(1),0:P
     OKEFH(1),20
6160 POKEW(1),33:FORU=1TO20:NEXTU,P:POKEW
     (1),0:GOSUB2010:POKEV+45,10
6170 PK=PEEK(V+30):GOTO680
6500 IFPEEK(V+5)<63THENRETURN
6510 POKEW(3),0:POKEFH(3),50:POKEW(3),17:
     Q=Q+100:PRINT"{HOME}{GRN}"TAB(6)Q:PO
     KEV+5,254
6520 PK=PEEK(V+31):RETURN
7000 POKEW(3),0:POKEFH(3),10:POKEW(3),129
     :FORP=PEEK(V+8)TOPEEK(V+8)+50STEP.5
7005 POKEV+8,P:NEXT
7010 IFPEEK(V+8)=230THEN10000
7020 PK=PEEK(V+31):GOSUB2000:GOTO530
8000 N=2:POKEW(1),0:POKEFH(1),5:POKEW(1),
     129:POKEW(1),128:POKEW(2),0:POKEW(2)
     ,17
8010 FORP=PEEK(V+1)TO250:POKEV+1,P:PK=PEE
     K(V):IFPK<255THENPK=PK+1
8020 POKEV,PK:POKEFH(2),255-P:FORU=1TO5:N
     EXTU,P:POKEW(2),0
8030 I=I-1:IFI=-1THEN10000
8040 GOSUB2000:POKEV,32:POKEV+1,162:POKEV
     +4,32:POKEV+5,254:POKEV+39,15:GOTO45
     0
8500 I=I-1:POKEW(2),0:POKEFH(2),10:POKEW(
     2),129
8510 FORP=1TO1000:NEXT:POKEW(2),0:IFI=-1T
     HEN10000
8520 PRINT"{HOME}{ 16 GIU'}[<7>]
     { 40 SPAZI}";
8530 POKEV+5,254:POKEV+1,164:POKE49799,0:
     PRINT"{HOME}"TAB(14)LI$(I)
8540 PK=PEEK(V+31):GOTO3100
10000 PRINT"{HOME}{ 12 GIU'}"SPC(15)"
      {WHT}GAME OVER";
10010 FORP=1TO3:POKEW(P),0:NEXT:POKEW(2),
      0:POKEW(2),33
10020 FORP=255TO50STEP-1:POKEFH(2),P:NEXT
      :POKEW(2),0
10030 FORP=1TO1000:NEXT:POKEV+21,0:PRINT"
      {CLR}"
10040 IFQ>HITHENHI=Q:GOSUB10100
10050 GOTO200
10100 HI$="":PRINT"{CLR}{ 10 GIU'}{RED}YO
      UR NAME IS... ";
10110 OPEN1,0:INPUT#1,A$:CLOSE1
10120 IFLEN(A$)>15THEN10100
10130 FORP=1TOLEN(A$):B$=MID$(A$,P,1)
10140 IF(B$<"A"ORB$>"Z")ANDB$<>" "THENP=L
      EN(A$)+1:GOTO10100
10150 NEXT:HI$=A$:GOTO200
11000 N=2:PK=PEEK(V+8):PK=230-PK:PK=PK/50
      :BO=500*PK:Q=Q+BO
11010 PRINT"{HOME} {BLU}BONUS"BO"PTS.
      {RED}SCORE"Q
11020 FORP=1TO1000:NEXT:FORP=1TO3:POKEW(P
      ),0:NEXT:FORP=1TO1000:NEXT
11030 I=I+1:IFI>2THENI=2
11040 SG=SG+1:POKEV+39,15:GOTO3000
11500 SG=SG+1:I=I+1:IFI>2THENI=2
11510 POKEW(2),0:POKEW(2),65:FORP=50TO250
      :POKEFH(2),P:NEXT:POKEW(2),0
11520 POKEV+21,0:GOTO400
```

1750-1953 - Definiscono gli sprite. Inoltre le linee 1940-1943 memorizzano da 49800 a 49831 le note della musichetta di sottofondo, nella forma Lo-Byte - Hi-Byte.
2000-2010 - Posizionano a caso la stella cadente e lo scrigno.
2050-2070 - Gestiscono la funzione del tasto "Fire" nei livelli dispari.
3000-3200 - Linee principali che controllano lo svolgimento dei livelli pari.
6000-6170 - Gestiscono le collisioni sprite-sprite nei livelli pari. In particolare 6050 - 6070 collisione cavaliere - stella cadente; 6100 - 6110 collisione cavaliere - freccia; 6150 - 6170 collisione cavaliere - scrigno.
6500-6520 - Collisione freccia - uccello nei livelli pari. Aggiornamento del punteggio.
7000-7020 - Movimento della nube verso la Luna.
8000-8040 - Routine di perdita di una vita nei livelli dispari. Fine del gioco se la scorta è esaurita.
8500-8540 - Come 8000 - 8040, nei livelli pari.
10000-11050 - Routine di fine partita.
10100-10150 - Se il punteggio memorizzato nella variabile Q è superiore al record registrato in HI, chie-

Novità firmate Jackson.





GRUPPO EDITORIALE JACKSON

La biblioteca che fa testo.

Giulio Carducci

LE APPLICAZIONI DEL COMPUTER NELL'UFFICIO MODERNO

Questo libro vuole essere un'introduzione all'informatica e, nel contempo, una guida all'utilizzo consapevole dello strumento del giorno, il personal computer, nell'ufficio moderno e nello studio professionale. È rivolto pertanto, ad un vasto pubblico: addetti ai vari settori dell'azienda, quadri, dirigenti, ingegneri, architetti, professionisti in genere.

Cod. 407H Pag. 132 Lire 23.000

Michael Browne

UNITÀ A DISCHI PER MICROCOMPUTER

Il libro, destinato a lettori con una buona conoscenza di base dell'uso di un calcolatore e del linguaggio BASIC, descrive il funzionamento dell'unità a dischi di un personal computer e il significato dei comandi relativi, con particolare attenzione per le diverse tecniche di gestione dei file su disco.
Il libro è arricchito di diversi programmi esempio in BASIC Commodore 4.0, relativi alle diverse tecniche di organizzazione e gestione dei file.

Cod. 300P Pag. 156 Lire 15.000

Mauro Salvemini

URBANISTICA E INFORMATICA

Sempre più vaste sono le applicazioni dell'informatica in ogni settore della vita e del lavoro, e in particolare quelle della computer grafica. Neanche mestieri e professioni con una origine antica come l'architetto o l'urbanistica si "salvano" dall'onda dell'informatica.
Un entusiasta utilizzatore di quest'area culturale racconta in questo libro per i colleghi e gli studenti di architettura e di urbanistica tutti i vantaggi e gli avanzamenti possibili nella progettazione e lo studio di nuovi edifici, complessi edilizi o urbani grazie alla computer grafica.

Cod. 801P Pag. 224 Lire 30.000

Graziella Tongoni

LA COMUNICAZIONE CAMBIATA

Questo libro vuole offrire una occasione a tutti i suoi lettori per analizzare, scoprire e potenziare le proprie capacità comunicative e divenire protagonisti reali di una ampia trasformazione in corso, di cui uno degli aspetti più evidenti è costituito dall'espandersi progressivo delle tecnologie.

Cod. 538P Pag. 112 Lire 10.500

Arthur Naiman

WORD STAR

Perché un computer, anche di categoria micro, sia utile nell'attività di ufficio è essenziale che permetta anche di effettuare l'elaborazione dei testi, con tutte le funzioni che ciò comporta. I prodotti software a ciò destinati sono oggi moltissimi ma solo alcuni emergono per completezza e flessibilità: fra questi Wordstar è sicuramente uno dei più riusciti.

Cod. 525P Pag. 222 Lire 23.000

---

ritagliare (o fotocopiare) e spedire in busta chiusa a:
GRUPPO EDITORIALE JACKSON - Divisione Libri - Via Rosellini, 12 - 20124 Milano

CEDOLA DI COMMISSIONE LIBRARIA

VOGLIATE SPEDIRMI

n° copie	codice	Prezzo unitario	Prezzo totale
		Totale	

☐ Pagherò contrassegno al postino il prezzo indicato più L. 3.000 per contributo fisso **spese di spedizione.**

Condizioni di pagamento con esenzione del contributo spese di spedizione:

☐ Allego assegno della Banca ____________

n° ____________

☐ Allego fotocopia del versamento su c/c n. 11666203 a voi intestato

☐ Allego fotocopia di versamento su vaglia postale a voi intestato

Nome

Cognome

Via

Cap Città Prov.

Data Firma

Spazio riservato alle Aziende. Si richiede l'emissione di fattura

Partita I.V.A. | | | | | | | | | | | |

ORDINE MINIMO L. 50.000

dono il nome del giocatore (HI) e aggiornano il record (HI=Q).
11000-11520 - Fanno avanzare il gioco al livello successivo a quello appena giocato.

La grafica

Il gioco si avvale di una interessante grafica in Multicolor. Quasi tutte le figure presenti sullo schermo, infatti, sono realizzate con questo metodo.
Nella grafica monocolore, ad ogni pixel corrisponde un bit: così un byte può definire ben 8 pixel. Nella grafica Multicolor, ad ogni pixel corrispondono due bit. Le combinazioni possibili ora non sono più due (0 e 1), ma ben 4: 00, 01, 10, 11. Se ad ognuna di queste combinazioni facciamo corrispondere un colore, vedremo che i colori disponibili sono quattro. Poichè nel modo multicolor i pixel sono di lunghezza doppia, ad un byte corrispondono quattro pixel raddoppiati.

Il suono

La musichetta di sottofondo è realizzata tramite una routine in linguaggio macchina, che legge le note memorizzate da 49800 a 49831 e le riporta, una per volta, sui registri dell'oscillatore 1. Altri effetti sonori sono ottenuti facendo variare molto velocemente il contenuto dei registri sonori. La qualità del sintetizzatore di cui è dotato il C 64 permetterebbe effetti di gran lunga migliori di quelli presenti in "Skywalker". Purtroppo la gestione del suono è molto complessa ed occupa molta memoria.

Alcune raccomandazioni

L'uso dei caratteri ridefinibili limita pesantemente la disponibilità di memoria. "Skywalker" sfrutta completamente la memoria disponibile. Perciò è molto importante che nel listato non vengano inseriti elementi estranei (si evitino, dunque, le Rem e gli spazi fra le diverse istruzioni).
Inoltre è molto importante copiare con attenzione le linee che contengono le routine in linguaggio macchina: anche un piccolo errore potrebbe determinare il blocco totale del sistema. Perciò è bene salvare il programma prima di dare il Run.
Non mi resta dunque che augurarvi buon divertimento con l'esecuzione di questo emozionante gioco di "movimento". ■

Zero e Croci con lo Spectrum

PERSONAL SOFTWARE

Un appassionante gioco di strategia

di *Anacleto Furlan*

Zero e croci è un gioco di strategia su scacchiera appartenente alla famiglia dei giochi di filetto. Le sue regole non sono complicate: su una scacchiera di 8x8 caselle due giocatori mettono, alternativamente le loro pedine finchè la riempiono, per due volte. Il gioco si svolge quindi in due manche: nella prima un giocatore assume il ruolo di Aggressore e l'altro di Difensore, nella seconda i ruoli si invertono. Obiettivo dell'Aggressore è fare in modo che si creino allineamenti più lunghi possibili di pedine uguali disposte consecutivamente: ad ogni allineamento (anche di pedine del Difensore!), formato da almeno tre pedine uguali, verrà corrisposto all'Aggressore un punteggio equivalente al numero di pedine meno due. Il Difensore, che non ottiene nessun punteggio, deve stare attento a non allineare le proprie pedine nel tentativo di interrompere gli allineamenti avversari. Vincerà la partita quel giocatore che otterrà il punteggio più alto.

Questo programma permette tre opzioni: l'utente può utilizzarlo per giocare con un'altra persona, per sfidare Spectrum, per osservare il gioco di Spectrum contro se stesso. La versione completa gira sullo Spectrum 48 Kbyte (listato 1). C'è anche

Listato 1. *Zero e croci per Spectrum 48 Kbyte.*

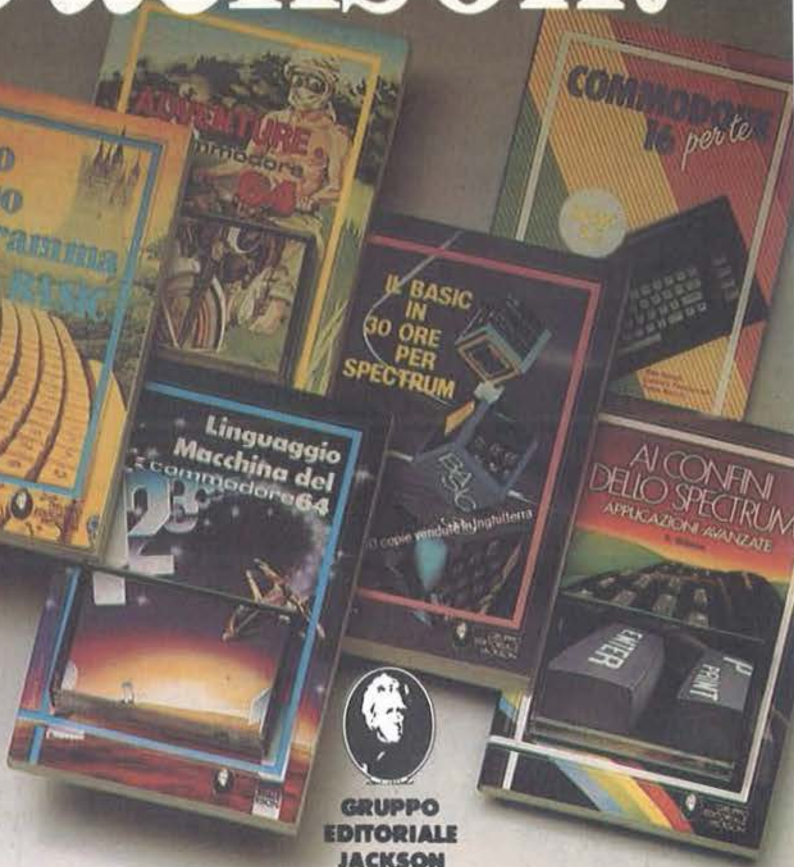
```
  10 REM ** ZERO E CROCI **
  20 REM © 1984 Anacleto Furlan
         Viale Papa Giovanni 35
         31015 Conegliano (TV)
  30 GO SUB 2625: GO SUB 1310
  40 GO SUB 2910: GO SUB 2710: G
O SUB 3000: GO TO 2000
 100 REM ricerca filetti di Spec
trum
 110 REM ricerca NS
 113 IF tot=35 THEN GO TO 194
 115 PRINT BRIGHT 1;AT 20,3;"S T
O   P E N S A N D O !"; BRIGHT
0;AT 21,11;"livello";" ";liv
 120 LET r=1: LET q=0
 130 FOR m=1 TO 6: FOR n=1 TO 3
 140 LET h$=d$(n,m)+d$(n+1,m)+d$
(n+2,m)+d$(n+3,m): GO SUB 305
 150 NEXT n: NEXT m
 160 REM ricerca EO
 170 LET r=0: LET q=1: FOR n=1 T
O 6: FOR m=1 TO 3
 180 LET h$=d$(n,m TO m+3): GO S
UB 305
 190 NEXT m: NEXT n: GO TO 210
 194 FOR n=1 TO 6: FOR m=1 TO 6:
 IF d$(n,m)=" " THEN LET a1=n: L
ET a2=m: GO TO 1480
 198 NEXT m: NEXT n
 200 REM ricerca NO-SE
 210 LET r=1: LET q=1
 220 FOR m=1 TO 3: FOR n=1 TO 3
 230 LET h$=d$(n,m)+d$(n+1,m+1)+
d$(n+2,m+2)+d$(n+3,m+3): GO SUB
305
 240 NEXT n: NEXT m
 250 REM ricerca NE-SO
 260 LET r=1: LET q=-1
 270 FOR m=4 TO 6: FOR n=1 TO 3
 280 LET h$=d$(n,m)+d$(n+1,m-1)+
d$(n+2,m-2)+d$(n+3,m-3): GO SUB
305
 290 NEXT n: NEXT m: IF liv<4 TH
EN GO TO 705
 295 GO TO 610
 300 REM punteggio coppie
 305 IF h$="    " THEN RETURN
 307 IF h$(1)<>" " AND h$(2)<>"
" AND h$(3)<>" " AND h$(4)<>" "
THEN RETURN
 310 LET giro=1
 315 LET o=1: LET p=3
 320 IF h$(o TO p)=" "+m$+m$ OR
h$(o TO p)=m$+" "+m$ OR h$(o TO
p)=m$+m$+" " THEN GO SUB 360
 330 IF o=1 THEN LET o=2: LET p=
4: GO TO 320
 340 LET m$=("O" AND m$="X")+("X
" AND m$="O"): LET giro=giro-2.5
: IF giro=-1.5 THEN GO TO 315
 342 IF dif=1 THEN GO TO 410
 344 IF h$=m$+m$+"  " THEN LET e
(n+3*r,m+3*q)=e(n+3*r,m+3*q)-300
0: GO TO 410
 347 IF h$="  "+m$+m$ THEN LET e
(n,m)=e(n,m)-3000
 348 IF liv=1 THEN RETURN
 350 GO TO 410
 360 FOR l=o-1 TO p-1: IF h$(l+1
)=" " THEN LET e(n+l*r,m+l*q)=e(
n+l*r,m+l*q)+dif*10000/giro: RET
URN
 370 NEXT l
 400 REM punteggio singoletti ce
ntrali
 410 LET giro=1
 420 IF h$=" "+m$+"  " OR h$="  "
+m$+" " THEN GO SUB 450
 430 LET m$=("O" AND m$="X")+("X
" AND m$="O"): LET giro=giro-2.5
: IF giro=-1.5 THEN GO TO 420
 435 IF liv=2 THEN RETURN
 440 GO TO 510
 450 FOR l=1 TO 2: IF h$(l+1)="
" THEN LET e(n+l*r,m+l*q)=e(n+l*
r,m+l*q)+dif*1000/giro: RETURN
 460 NEXT l
 500 REM punteggio singoletti es
tremali
```

Seguito listato versione 48 Kbyte.

```
 510 LET giro=1
 520 IF h$=m$+"  " OR h$="  "+
m$ THEN GO SUB 550
 530 LET m$=("O" AND m$="X")+("X
" AND m$="O"): LET giro=giro-2.5
: IF giro=-1.5 THEN GO TO 520
 540 RETURN
 550 LET e(n+r,m+q)=e(n+r,m+q)+d
if*100/giro: LET e(n+2*r,m+2*q)=
e(n+2*r,m+2*q)+dif*100/giro: RET
URN
 600 REM ricerca agli angoli
 610 LET m=-1: LET n=1: LET l=1:
 GO SUB 650
 620 LET m=1: GO SUB 650
 630 LET n=6: LET l=-1: GO SUB 6
50
 640 LET m=-1: GO SUB 650: GO TO
 705
 650 LET h$=d$(3,n)+d$(3+m,n+l)+
d$(3+2*m,n+2*l): FOR u=1 TO 3: I
F h$(u)<>" " THEN NEXT u: RETURN

 660 LET giro=1
 670 LET z$=h$(1 TO u-1)+h$(u+1
TO 3): IF z$=m$+m$ THEN LET e(3+
(u-1)*m,n+(u-1)*l)=e(3+(u-1)*m,n
+(u-1)*l)+dif*10000/giro
 680 LET m$=("O" AND m$="X")+("X
" AND m$="O"): LET giro=giro-2.5
: IF giro=-1.5 THEN GO TO 670
 690 RETURN
 700 REM punteggio probabilita'
di filetto
 705 RESTORE 710: FOR n=1 TO 6:
FOR m=1 TO 6: READ a
 710 DATA 18,17,19,19,17,18
 720 DATA 17,21,21,21,21,17
 730 DATA 19,21,23,23,21,19
 740 DATA 19,21,23,23,21,19
 750 DATA 17,21,21,21,21,17
 760 DATA 18,17,19,19,17,18
 770 IF d$(n,m)=" " THEN LET e(n
,m)=e(n,m)+dif*a: GO TO 790
 780 LET e(n,m)=-100000
 790 NEXT m: NEXT n
 800 REM scelta punteggio max e
indirizzamento a visualizzazione
 810 DIM k(36): FOR m=1 TO 6: FO
R l=1 TO 6: LET k((m-1)*6+l)=e(m
,l): NEXT l: NEXT m
 820 LET lax=0: FOR n=1 TO 35
 830 IF k(n+1)>k(n-lax) THEN LET
 k(n-lax)=-100000: GO TO 850
 840 GO TO 870
 850 IF lax>0 THEN GO SUB 980
 860 NEXT n: GO TO 890
 870 IF k(n+1)=k(n-lax) THEN LET
 lax=lax+1: NEXT n: GO TO 890
 880 LET k(n+1)=-100000: LET lax
=lax+1: NEXT n
 890 LET z=0: FOR n=1 TO 36
 900 IF k(n)>-100000 THEN LET z=
z+1: LET w=n
 910 NEXT n: LET j=0
 920 IF z=1 THEN GO TO 950
 930 LET nmax=1+INT (z*RND): FOR
 w=1 TO 36: IF k(w)=-100000 THEN
 NEXT w
 940 LET j=j+1: IF j<nmax THEN N
EXT w
 950 LET a1=1: IF w<7 THEN LET a
2=w: GO TO 1480
 960 LET a1=a1+1: LET w=w-6: IF
w<7 THEN LET a2=w: GO TO 1480
 970 GO TO 960
 980 FOR w=1 TO n: IF k(w)<>-100
000 THEN LET k(w)=-100000
 990 NEXT w: LET lax=0: RETURN
1000 REM punteggio aggressore
1010 REM filetti verticali
1020 LET aggr=0: IF tot<4 THEN R
ETURN
1025 FOR m=1 TO 6: FOR n=1 TO 4
1030 LET h$=d$(n,m)+d$(n+1,m)+d$
(n+2,m): GO SUB 1210
1040 NEXT n: NEXT m
1050 REM filetti orizzontali
1060 FOR n=1 TO 6: FOR m=1 TO 4
1070 LET h$=d$(n,m TO m+2): GO S
UB 1210
1080 NEXT m: NEXT n
1100 REM filetti NO-SE
1110 FOR m=1 TO 4: FOR n=1 TO 4
1120 LET h$=d$(n,m)+d$(n+1,m+1)+
d$(n+2,m+2): GO SUB 1210
1130 NEXT n: NEXT m
1150 REM filetti NE-SO
1160 FOR m=3 TO 6: FOR n=1 TO 4
1170 LET h$=d$(n,m)+d$(n+1,m-1)+
d$(n+2,m-2): GO SUB 1210
1180 NEXT n: NEXT m
1190 PRINT AT 12,11;aggr: RETURN
1200 REM assegnazione punteggio
aggressore
1210 IF h$="XXX" OR h$="OOO" THE
N LET aggr=aggr+1
1220 RETURN
1300 REM caratteri grafici
1310 RESTORE 1330: FOR n=0 TO 7:
 READ l: POKE USR "a"+n,l: READ
l: POKE USR "b"+n,l: NEXT n
1320 RETURN
1330 DATA 1,255,3,255,7,128,15,2
48
1340 DATA 31,248,48,48,96,24,0,0
1400 REM input mossa
1410 LET fl=(0 AND fl=1)+(1 AND
fl=0): PRINT AT 9,0;"Mette "; F
LASH 1;("AGG." AND fl=0)+("DIF."
 AND fl=1); FLASH 0;" "; FLASH 1
;m$: BEEP .3,20
1413 IF tot=35 THEN GO TO 194
1415 GO TO 1425
1420 BEEP .5,-25
1425 INPUT a$: IF LEN a$<>2 THEN
 GO TO 1420
1430 FOR n=1 TO 6: IF a$(1)=b$(n
) THEN LET a2=n: GO TO 1450
1440 NEXT n: GO TO 1420
1450 FOR n=1 TO 6: IF a$(2)=c$(n
) THEN LET a1=n: GO TO 1470
1460 NEXT n: GO TO 1420
1470 IF d$(a1,a2)<>" " THEN GO T
O 1420
1480 IF sg=1 THEN PRINT AT 20,3;
"                              ";AT
21,11;"            ": LET m$=w$
1490 IF n$="" OR tot=0 THEN GO T
O 1510
1500 PRINT INK (1 AND m$="O")+(2
 AND m$="X");AT 2*a3+3,2*a4+15;n
$
1510 LET d$(a1,a2)=m$: LET a3=a1
: LET a4=a2: LET n$=m$
1520 PRINT INK (1 AND m$="X")+(2
 AND m$="O"); FLASH 1;AT 2*a3+3,
2*a4+15;m$
1530 BEEP .2,10: PRINT AT 9,6;"?
     ";AT 10,6;"          ": GO SU
B 1020
1540 REM contatore pedine
1550 LET m$=("X" AND m$="O")+("O
" AND m$="X"): IF sp=1 THEN LET
sg=(0 AND sg=1)+(1 AND sg=0)
1560 LET tot=tot+1: IF tot=36 TH
EN GO TO 2810
1570 IF NOT sg THEN PRINT AT 9,7
;"    ";AT 10,6;"          ": GO
TO 1410
1580 LET fl=(0 AND fl=1)+(1 AND
fl=0): GO TO 1620
1600 REM mossa a Spectrum
1610 LET w$="X": GO SUB 2950: LE
T primo=1
1620 IF sp=2 THEN LET dif=(1 AND
 dif=-1)+(-1 AND dif=1)
1630 PRINT AT 9,0;"Mette "; FLAS
```

Novità Jackson.



David Lawrence

LINGUAGGIO MACCHINA DEL COMMODORE 64

Il libro apre nuovi orizzonti a tutti coloro che sono interessati alla programmazione in linguaggio macchina del COMMODORE 64.

Con cassetta
Cod. 572D Pag. 208 Lire 29.000

Clive Prigmore

IL BASIC IN 30 ORE PER SPECTRUM

Questo semplice corso di autoistruzione insegna a programmare, e un programma ha sempre bisogno di due ingredienti, un linguaggio e una struttura: dunque questo libro non insegna solo il BASIC, ma anche come si organizza correttamente un buon programma.

Cod. 501B Pag. 360 Lire 40.000

Rodnay Zaks

IL TUO PRIMO PROGRAMMA IN BASIC

La diffusione del BASIC per la sua semplicità e quasi "naturalezza" di programmazione fa sì che una cultura generale sull'informatica e la sua applicazione non può prescindere da una conoscenza di base di questo linguaggio. Questo lo scopo del libro: permettere anche a chi ha soltanto una cultura di base, di capire che cos'è il BASIC e come si usa.

Cod. 507B Pag. 216 Lire 19.500

Czes Kosniowski

MATEMATICA E COMMODORE 64

Tutte le funzioni matematiche disponibili sul C64 sono qui descritte, ed il loro uso è illustrato con programmi che possono essere utilizzati dal lettore all'interno dei suoi, per particolari applicazioni.
Il libro contiene anche informazioni e programmi su altri argomenti, come i codici e la crittografia, i numeri casuali, le serie, la trigonometria, i numeri primi e l'analisi statistica dei dati.

Con cassetta
Cod. 570D Pag. 160 Lire 24.000

... Williams

AI CONFINI DELLO SPECTRUM
Applicazioni avanzate

Un esame attento dei listati consentirà al lettore di apprendere i "segreti" della programmazione strutturata e migliorare notevolmente le proprie capacità di programmatore.
I programmi presentati vanno dagli arcade più famosi, tra cui il celebre "Spectrum Invaders" ai programmi di utilità più interessanti, dai giochi d'azzardo ai programmi didattici, dai programmi funzionali a quelli di matematica e di giochi di strategia.

Con cassetta
Cod. 414B Pag. 180 Lire 28.000

Mike Grace

ADVENTURE E COMMODORE 64

Un manuale per ideare e utilizzare programmi di Adventure basati esclusivamente sul testo.
Una progettazione modulare del programma rende più facile la comprensione della struttura. L'abilità nella programmazione avrà modo di emergere durante la costruzione graduale dell'Adventure, per modificare il programma dimostrativo, o addirittura per costruirne uno nuovo.

Con cassetta
Cod. 571D Pag. 240 Lire 35.000

Rita Bonelli

COMMODORE 16 PER TE: BASIC 3.5

È un libro di introduzione al BASIC C16 con il classico taglio didattico Bonelli. Adatto per la Scuola media inferiore e per chi non conosce l'informatica.
La cassetta allegata al libro contiene diverse lezioni, una per ogni capitolo, che devono essere lette prima del capitolo relativo.

Con cassetta
Cod. 413B Pag. 296 Lire 35.000



GRUPPO EDITORIALE JACKSON

La biblioteca che fa testo.

ritagliare (o fotocopiare) e spedire in busta chiusa a:
GRUPPO EDITORIALE JACKSON – Divisione Libri - Via Rosellini, 12 - 20124 Milano

CEDOLA DI COMMISSIONE LIBRARIA

VOGLIATE SPEDIRMI

n° copie	codice	Prezzo unitario	Prezzo totale
		Totale	

☐ Pagherò contrassegno al postino il prezzo indicato più **L. 3.000 per contributo fisso spese di spedizione.**

Condizioni di pagamento con esenzione del contributo spese di spedizione:

☐ Allego assegno della Banca ____________________

n° ____________________

☐ Allego fotocopia del versamento su c/c n. 11666203 a voi intestato

☐ Allego fotocopia di versamento su vaglia postale a voi intestato

Nome

Cognome

Via

Cap Città Prov.

Data Firma

Spazio riservato alle Aziende. Si richiede l'emissione di fattura

Partita I.V.A. | | | | | | | | | | | |

ORDINE MINIMO L. 50.000

PERSONAL SOFTWARE

Zero e Croci con lo Spectrum

Seguito listato versione 48 Kbyte.

```
H 1;("AGGR." AND dif=1)+("DIF."
AND dif=-1);AT 10,6;"(Spec.)": I
F sp<>2 THEN BEEP .3,20: GO TO 1
660
1640 PRINT FLASH 1;AT 9,12;m$
1650 LET w$=m$
1660 IF primo=1 THEN GO SUB 1680
: LET a1=a5: GO SUB 1680: LET a2
=a5: LET primo=0: GO TO 1480
1670 DIM e(6,6): GO TO 110
1680 LET n=INT (RND*10): LET a5=
(3+INT (RND*2) AND n<7)+(2+INT (
RND*4) AND n>=7): RETURN
2000 REM menu'
2010 PRINT AT 8,0;"Premi-r- per"
'"le regole"'"del gioco"
2020 PRINT AT 17,0;"#############
##"'"      Premi:
   "
2030 PRINT AT 19,0;"-a- per gioc
are contro un amico "
2040 PRINT "-s- per giocare cont
ro Spectrum "
2050 PRINT "-p- per Spectrum con
tro Spectrum"
2100 REM indirizzamento
2110 PAUSE 0: IF INKEY$="r" THEN
 GO SUB 2310: GO TO 2340
2120 IF INKEY$="a" THEN LET sp=0
: LET sg=0: GO SUB 2310: GO SUB
2190: GO TO 2160
2130 IF INKEY$="s" THEN LET sp=1
: GO SUB 2310: GO SUB 2190: GO T
O 2210
2140 IF INKEY$="p" THEN LET sp=2
: LET sg=1: GO SUB 2310: GO SUB
2190: LET dif=-1: LET liv=4: GO
TO 1610
2150 GO TO 2110
2160 PRINT FLASH 1;AT 19,0;"Indi
co con X l'aggressore? (s/n)"';
FLASH 0;"(che mettera' per primo
)": PAUSE 0: IF INKEY$<>"s" THEN
 LET m$="O"
2170 GO SUB 2190: GO SUB 2950: G
O TO 1410
2180 REM cancella in basso
2190 FOR m=0 TO 31 STEP 4: FOR n
=18 TO 21: PAUSE 3: PRINT AT n,m
;"    ": NEXT n: NEXT m: RETURN
2200 REM opzioni quando gioca Sp
ectrum
2210 PRINT FLASH 1;AT 19,2;"SCEG
LI IL LIVELLO: ";AT 19,22;"1";AT
 19,24;"2";AT 19,26;"3";AT 19,28
;"4": PAUSE 0
2213 IF CODE INKEY$<49 OR CODE I
NKEY$>52 THEN GO TO 2210
2216 LET liv=VAL INKEY$: PRINT A
T 19,31;liv
2218 PRINT FLASH 1;AT 20,0;"CHE
SIMBOLO SCEGLI ? (X/O)    ": PAUS
E 0
2220 IF INKEY$="x" THEN LET m$="
X": GO TO 2250
2230 IF INKEY$="o" THEN LET m$="
O": GO TO 2250
2240 GO TO 2220
2250 LET w$=("X" AND m$="O")+("O
" AND m$="X"): PRINT AT 20,31;m$
2260 PRINT FLASH 1;AT 21,0;"VUOI
 INIZIARE DA AGGRESSORE ?": PAUS
E 0
2270 IF INKEY$="s" THEN PRINT AT
 21,31;"S": GO SUB 2190: LET sg=
0: LET primo=0: LET dif=-1: GO S
UB 2950: GO TO 1410
2280 PRINT AT 21,31;"N": GO SUB
2190: LET sg=1: LET primo=1: GO
SUB 2950: LET dif=1: LET fl=0: G
O TO 1620
2300 REM cancella centro
2310 FOR m=0 TO 11 STEP 4: FOR n
=8 TO 11: PAUSE 3: PRINT AT n,m;
"    ": NEXT n: NEXT m
2320 RETURN
2330 REM istruzioni
2340 LET j=2370: LET i=18: RESTO
RE j
2350 IF j=2560 THEN PAUSE 100: B
EEP .5,-20: GO TO 2010
2360 READ i$
2370 DATA "           REGOLE DEL GI
OCO"
2380 DATA "
   "
2390 DATA "Ogni partita si svolg
e con una"
2400 DATA "ANDATA e un RITORNO.
Nell'ANDATA"
2410 DATA "un giocatore  fa  da
 AGGRESSORE"
2420 DATA "e l'altro da DIFENSOR
E.  Al  RI-"
2430 DATA "TORNO  le  parti  si
 invertono."
2440 DATA "Scopo dell'AGGRESSORE
 e' realiz-"
2450 DATA "zare allineamenti lun
ghi il piu'"
2460 DATA "possibile: ogni allin
eamento var"
2470 DATA "ra' tanti punti  quan
ti  simboli"
2480 DATA "contiene, meno due.
   "
```

una versione per lo Spectrum 16 Kbyte, presentata nel listato 2, ottenuta dalla precedente eliminando i "fronzoli", ma anche alcune routine riguardanti la ricerca della migliore mossa quando gioca Spectrum.

Ciononostante alcune linee sono state "compattate" senza determinare tuttavia alcun ritardo nel tempo di "riflessione" dello Spectrum, che risulta essere in media di un minuto primo per ogni mossa. Per la versione completa invece il tempo è di circa un minuto e venti secondi per il livello più alto (4°), mentre agli altri livelli risulta intermedio fra i due valori citati.

Spectrum arbitro

Qualunque opzione di gioco l'utente scelga, Spectrum dirige il gioco: assegna la mossa a chi spetta (ricordandone il ruolo), ne controlla la correttezza, segna sulla scacchiera la pedina alla posizione voluta (linee 1410-1530) e, soprattutto, calcola la lunghezza di ogni allineamento, assegnando così il punteggio all'Aggressore (linee 1000-1220): infine dichiara l'esito della partita.

Sfida a Spectrum

E' decisamente l'opzione più interessante. Si può cercare di battere Spectrum a quattro diversi livelli di gioco. Il primo livello (il più debole) corrisponde circa al gioco della versione ridotta per 16 Kbyte del pro-

Zero e Croci con lo Spectrum

```
2490 DATA "Scopo del DIFENSORE
e' impedire"
2500 DATA "cio', evitando quin
di di al-"
2510 DATA "lineare anche i propr
i simboli,"
2520 DATA "per non regalare pun
ti all'AG-"
2530 DATA "GRESSORE (che e' il s
olo a se-"
2540 DATA "gnare punti).
"
2550 DATA "La manche termina qua
ndo il qua-"
2560 DATA "drato e' coperto di s
imboli. Al"
2570 DATA "termine del RITORNO i
 giocatori"
2580 DATA "confrontano i puntegg
i ottenuti."
2590 PRINT AT i,0;i$: GO SUB 260
0: GO TO 2360
2600 LET i=i+1: IF i=22 THEN LET
 i=18: PAUSE 150: BEEP .05,10: L
ET j=j+10: RESTORE j: GO TO 2350
2610 RETURN
2620 REM titoli
2630 PAPER 4: BORDER 4: INK 0: C
LS : FOR n=PI TO 0 STEP -PI/10
2640 BEEP .05,-10: PLOT 128,88:
DRAW 87*SIN n,87*COS n: PLOT 128
,88: DRAW -87*SIN n,-87*COS n
2650 NEXT n
2660 FOR n=2*PI TO 0 STEP -PI/10
2670 BEEP .05,10: CIRCLE 128,88,
n*87/(2*PI)
2680 PRINT BRIGHT 1;AT 9,10;"ZER
O E CROCI"; BRIGHT 0;AT 11,10;"d
i A. Furlan"
2690 NEXT n: BEEP 2,-15: PAPER 7
: BORDER 7: CLS : RETURN
2700 REM inizializzazioni variab
ili
2710 LET n=0: LET m=0: LET l=0
2720 LET b$="abcdef"
2730 LET c$="12345678"
2740 LET m$="X": LET n$=""
2750 DIM d$(6,6): LET aggr=0: LE
T dif=0
2760 LET a3=0: LET a4=0: LET fl=
1
2770 LET tot=0: LET manche=0
2780 RETURN
2800 REM termine manches
```

```
2810 LET manche=manche+1: LET to
t=0
2815 IF manche=1 AND sp=1 THEN L
ET dif=(-1 AND dif=1)+(1 AND dif
=-1): LET fl=1
2820 IF manche=1 THEN LET avv=ag
gr: DIM d$(6,6): BEEP 2,30: CLS
: GO SUB 2910: GO SUB 2950: GO S
UB 3010: LET tot=-1: PRINT AT 17
,0;"##############": LET primo=(
1 AND (dif=1 OR sp=2)): GO TO 15
50
2830 IF avv<aggr THEN BEEP 1,-10
: PRINT FLASH 1;AT 19,10;"VINCE
";m$;AT 20,7;"differenza ";aggr-
avv: GO TO 2860
2840 IF avv>aggr THEN BEEP 1,-10
: PRINT FLASH 1;AT 19,10;"VINCE
";("X" AND m$="O")+("O" AND m$="
X");AT 20,7;"differenza ";avv-ag
gr: GO TO 2860
2850 BEEP 1,0: PRINT FLASH 1;AT
19,10;"PARITA'!"
2860 PRINT AT 9,0;"       ";#1;"
PER CONTINUARE PREMI UN TASTO"
2870 PAUSE 0: CLEAR : GO TO 40
2900 REM indicazioni alte
2910 PRINT INVERSE 1;AT 1,10;"ZE
RO E CROCI"; INVERSE 0;AT 1,28;"
©◢"
2920 PRINT AT 4,0;"#############
#"
2930 RETURN
2940 REM indicazioni medie
2950 PRINT AT 7,3;("ANDATA" AND
manche=0)+("RITORNO" AND manche=
1);AT 12,0;"PUNTEGGIO:": IF manc
he=1 THEN PRINT AT 14,0;"PUNTEGG
IO";'"ANDATA:"; FLASH 1;AT 15,7;
aggr
2960 RETURN
3000 REM scacchiera
3010 PRINT AT 4,16;"▛";AT 16,16;
"▙";AT 4,28;"▜";AT 16,28;"▟"
3020 FOR n=17 TO 27: PRINT AT 4,
n;"▀";AT 16,n;"▄": NEXT n
3030 FOR n=5 TO 15: PRINT AT n,1
6;"▌";AT n,28;"▐": NEXT n
3040 LET m=124
3050 FOR n=0 TO 64 STEP 16: PLOT
 131,m-n: DRAW 97,0: NEXT n: IF
m=123 THEN GO TO 3070
3060 LET m=123: GO TO 3050
3070 LET m=147
3080 FOR n=0 TO 64 STEP 16: PLOT
 m+n,43: DRAW 0,97: NEXT n: IF m
=148 THEN GO TO 3100
3090 LET m=148: GO TO 3080
3100 PLOT 132,44: DRAW 95,0: DRA
W 0,95: DRAW -95,0: DRAW 0,-95
3110 FOR n=5 TO 15 STEP 2: PRINT
 AT n,29;n/2-1.5: NEXT n
3120 PRINT AT 17,17;"A B C D E F
"
3130 RETURN
```

gramma.

La routine che provvede alla scelta della migliore mossa (per Spectrum sia Aggressore che Difensore) va dalla linea 100 alla linea 999. Ad ogni casella della scacchiera viene assegnato un punteggio (linee 100-795), le rimanenti linee della routine determinano la posizione della casella a punteggio massimo.

Soffermiamoci ora sul primo gruppo di linee (100-795).

In quelle che vanno da 100 a 690 si assegna un punteggio in base alla presenza di altre pedine, nelle rimanenti si tiene conto solamente della probabilità per ogni casella di essere perno di filetti (per esempio le caselle centrali sono favorite). Dato che tale probabilità non è determinante, il punteggio assegnato da questa routine è basso rispetto a quelli assegnati in 100-795. Nelle linee comprese fra 100 e 550 vengono assegnati punteggi alle caselle analizzando la struttura di una stringa, h$, formata da quattro caselle consecutive. Le linee 110-290 provvedono a formare tutte le possibili strisce h$ e le indirizzano poi alle subroutine 300, 400, 500.

Nella 300 si studia h$ cercandovi la presenza di due pedine uguali che possono essere consecutive con a lato una casella vuota oppure separate da questa. Al punteggio di ogni casella, la cui occupazione permette di formare una terna di pedine allineate consecutivamente, viene aggiunto un punteggio molto alto se si forma un filetto dell'Aggressore, molto basso (negativo) se si forma un filet-

PERSONAL SOFTWARE

Zero e Croci con lo Spectrum

Seguito listato versione 48 Kbyte.

```
  10 REM ** ZERO E CROCI **(16K)
  20 REM © 1984 Anacleto Furlan
         Viale Papa Giovanni 35
         31015 Conegliano (TV)
  40 GO SUB VAL "2910": GO SUB V
AL "2710": GO SUB VAL "3000": GO
 TO VAL "2000"
 115 PRINT BRIGHT VAL "1";AT VAL
"20",VAL "3";"S T O     P E N S
A N D O !"
 120 LET r=1: LET q=0
 130 FOR m=1 TO 6: FOR n=1 TO 3
 140 LET h$=d$(n,m)+d$(n+1,m)+d$
(n+2,m)+d$(n+3,m): GO SUB 305
 150 NEXT n: NEXT m
 170 LET r=0: LET q=1: FOR n=1 T
O 6: FOR m=1 TO 3
 180 LET h$=d$(n,m TO m+3): GO S
UB 305
 190 NEXT m: NEXT n
 210 LET r=1: LET q=1
 220 FOR m=1 TO 3: FOR n=1 TO 3
 230 LET h$=d$(n,m)+d$(n+1,m+1)+
d$(n+2,m+2)+d$(n+3,m+3): GO SUB
305
 240 NEXT n: NEXT m
 260 LET r=1: LET q=-1
 270 FOR m=4 TO 6: FOR n=1 TO 3
 280 LET h$=d$(n,m)+d$(n+1,m-1)+
d$(n+2,m-2)+d$(n+3,m-3): GO SUB
305
 290 NEXT n: NEXT m: GO TO 610
 305 IF h$="    " THEN RETURN
 307 IF h$(1)<>" " AND h$(2)<>"
" AND h$(3)<>" " AND h$(4)<>" "
THEN RETURN
 310 LET giro=1
 315 LET o=1: LET p=3
 320 IF h$(o TO p)=" "+m$+m$ OR
h$(o TO p)=m$+" "+m$ OR h$(o TO
p)=m$+m$+" " THEN GO SUB 360
 330 IF o=1 THEN LET o=2: LET p=
4: GO TO 320
 340 LET m$=("O" AND m$="X")+("X
" AND m$="O"): LET giro=giro-2.5
: IF giro=-1.5 THEN GO TO 315
 350 GO TO 410
 360 FOR l=o-1 TO p-1: IF h$(l+1
)=" " THEN LET e(n+l*r,m+l*q)=e(
n+l*r,m+l*q)+dif*10000/giro: RET
URN
 370 NEXT l
 410 LET giro=1
 420 IF h$="  "+m$+" " OR h$=" "
+m$+"  " THEN GO SUB 450
 430 LET m$=("O" AND m$="X")+("X
" AND m$="O"): LET giro=giro-2.5
: IF giro=-1.5 THEN GO TO 420
 440 RETURN
 450 FOR l=1 TO 2: IF h$(l+1)="
" THEN LET e(n+l*r,m+l*q)=e(n+l*
r,m+l*q)+dif*1000/giro: RETURN
 460 NEXT l
 710 FOR n=1 TO 6: FOR m=1 TO 6
 720 IF d$(n,m)=" " THEN GO TO 7
40
 730 LET e(n,m)=-100000
 740 NEXT m: NEXT n
 810 DIM k(36): FOR m=1 TO 6: FO
R l=1 TO 6: LET k((m-1)*6+l)=e(m
,l): NEXT l: NEXT m
 820 LET lax=0: FOR n=1 TO 35
 830 IF k(n+1)>k(n-lax) THEN LET
 k(n-lax)=-100000: GO TO 850
 840 GO TO 870
 850 IF lax>0 THEN GO SUB 980
 860 NEXT n: GO TO 890
 870 IF k(n+1)=k(n-lax) THEN LET
 lax=lax+1: NEXT n: GO TO 890
 880 LET k(n+1)=-100000: LET lax
=lax+1: NEXT n
 890 LET z=0: FOR n=1 TO 36
 900 IF k(n)>-100000 THEN LET z=
z+1: LET w=n
 910 NEXT n: LET j=PI-PI
 920 IF z=1 THEN GO TO 950
 930 LET nmax=1+INT (z*RND): FOR
w=1 TO 36: IF k(w)=-100000 THEN
 NEXT w
 940 LET j=j+1: IF j<nmax THEN N
EXT w
 950 LET a1=1: IF w<7 THEN LET a
2=w: GO TO 1480
 960 LET a1=a1+1: LET w=w-6: IF
w<7 THEN LET a2=w: GO TO 1480
 970 GO TO 960
 980 FOR w=1 TO n: IF k(w)<>-100
000 THEN LET k(w)=-100000
 990 NEXT w: LET lax=0: RETURN
1020 LET aggr=0: IF tot<4 THEN R
ETURN
1025 FOR m=1 TO 6: FOR n=1 TO 4
1030 LET h$=d$(n,m)+d$(n+1,m)+d$
(n+2,m): GO SUB 1210
1040 NEXT n: NEXT m
1060 FOR n=1 TO 6: FOR m=1 TO 4
1070 LET h$=d$(n,m TO m+2): GO S
UB 1210
1080 NEXT m: NEXT n
1110 FOR m=1 TO 4: FOR n=1 TO 4
1120 LET h$=d$(n,m)+d$(n+1,m+1)+
d$(n+2,m+2): GO SUB 1210
1130 NEXT n: NEXT m
1160 FOR m=3 TO 6: FOR n=1 TO 4
1170 LET h$=d$(n,m)+d$(n+1,m-1)+
d$(n+2,m-2): GO SUB 1210
1180 NEXT n: NEXT m
1190 PRINT AT 12,11;aggr: RETURN

1210 IF h$="XXX" OR h$="OOO" THE
N LET aggr=aggr+1
1220 RETURN
1410 LET fl=(0 AND fl=1)+(1 AND
fl=0): PRINT AT 9,0;"Mette  "; F
LASH 1;("AGG." AND fl=0)+("DIF."
 AND fl=1); FLASH 0;" "; FLASH 1
;m$: BEEP .3,20: GO TO 1425
1420 BEEP .5,-25
1425 INPUT a$: IF LEN a$<>VAL "2
" THEN GO TO 1420
1430 FOR n=1 TO 6: IF a$(1)=b$(n
) THEN LET a2=n: GO TO 1450
1440 NEXT n: GO TO VAL "1420"
1450 FOR n=1 TO 6: IF a$(2)=c$(n
) THEN LET a1=n: GO TO 1470
1460 NEXT n: GO TO VAL "1420"
1470 IF d$(a1,a2)<>" " THEN GO T
O VAL "1420"
1480 IF sg=VAL "1" THEN PRINT AT
 20,3;"
 ": LET m$=w$
1490 IF n$="" OR tot=PI-PI THEN
GO TO VAL "1510"
1500 PRINT INK (1 AND m$="O")+(2
 AND m$="X");AT 2*a3+3,2*a4+15;n
$
1510 LET d$(a1,a2)=m$: LET a3=a1
: LET a4=a2: LET n$=m$
1520 PRINT INK (1 AND m$="X")+(2
 AND m$="O"); FLASH 1;AT 2*a3+3,
2*a4+15;m$
1530 BEEP .2,10: PRINT AT 9,6;"?
   ";AT 10,6;"      ": GO SU
B 1020
1550 LET m$=("X" AND m$="O")+("O
" AND m$="X"): IF sp=1 THEN LET
sg=(0 AND sg=1)+(1 AND sg=0)
1560 LET tot=tot+1: IF tot=36 TH
```

Zero e Croci con lo Spectrum

Seguito listato versione 16 Kbyte.

```
EN GO TO 2810
1570 IF NOT sg THEN PRINT AT 9,7
;"      ";AT 10,6;"    ": GO
TO 1410
1580 LET fl=(0 AND fl=1)+(1 AND
fl=0): GO TO VAL "1620"
1610 LET w$="X": GO SUB VAL "295
0": LET primo=VAL "1"
1620 IF sp=VAL "2" THEN LET dif=
(VAL "1" AND dif=VAL "-1")+(VAL
"-1" AND dif=VAL "1")
1630 PRINT AT 9,0;"Mette "; FLAS
H 1;("AGGR." AND dif=1)+("DIF."
AND dif=-1);AT 10,6;"(Spec.)": I
F sp<>VAL "2" THEN BEEP .3,20: G
O TO 1660
1640 PRINT FLASH VAL "1";AT VAL
"9",VAL "12";m$
1650 LET w$=m$
1660 IF primo=1 THEN LET a1=3+IN
T (RND*4): LET a2=3+INT (RND*4):
 LET primo=PI-PI: GO TO VAL "148
0"
1670 DIM e(VAL "6",VAL "6"): GO
TO VAL "110"
2020 PRINT AT 17,0;"############
##"'"     Premi:"
2030 PRINT AT VAL "19",PI-PI;"-a
- per giocare contro un amico "
2040 PRINT "-s- per giocare cont
ro Spectrum "
2050 PRINT "-p- per Spectrum con
tro Spectrum"
2110 PAUSE PI-PI: IF INKEY$="i"
THEN GO SUB VAL "2310": GO TO VA
L "2340"
2120 IF INKEY$="a" THEN LET sp=P
I-PI: LET sg=PI-PI: GO SUB VAL "
2190": GO TO VAL "2160"
2130 IF INKEY$="s" THEN LET sp=V
AL "1": GO SUB VAL "2190": GO TO
 VAL "2210"
2140 IF INKEY$="p" THEN LET sp=2
: LET sg=1: GO SUB 2190: LET dif
=-1: GO TO 1610
2150 GO TO VAL "2110"
2160 PRINT FLASH 1;AT 19,PI-PI;"
Indico con X l'aggressore? (s/n)
"; FLASH 0;"(che mettera' per p
rimo)": PAUSE 0: IF INKEY$<>"s"
THEN LET m$="O"
2170 GO SUB VAL "2190": GO SUB V
AL "2950": GO TO VAL "1410"
2190 FOR m=0 TO 31 STEP 4: FOR n
=18 TO 21: PAUSE 3: PRINT AT n,m
;"    ": NEXT n: NEXT m: RETURN
2210 PRINT FLASH VAL "1";AT VAL
"19",PI-PI;"CHE SIMBOLO SCEGLI ?
 (X/O)": PAUSE PI-PI
2220 IF INKEY$="x" THEN LET m$="
X": GO TO VAL "2250"
2230 IF INKEY$="o" THEN LET m$="
O": GO TO VAL "2250"
2240 GO TO VAL "2220"
2250 LET w$=("X" AND m$="O")+("O
" AND m$="X"): PRINT AT VAL "19"
,VAL "31";m$
2260 PRINT FLASH VAL "1";AT VAL
"20",PI-PI;"VUOI INIZIARE DA AGG
RESSORE ?": PAUSE PI-PI
2270 IF INKEY$="s" THEN PRINT AT
 VAL "20",VAL "31";"S": GO SUB V
AL "2190": LET sg=0: LET primo=0
: LET dif=-1: GO SUB 2950: GO TO
 1410
2280 PRINT AT 20,31;"N": GO SUB
2190: LET sg=VAL "1": LET primo=
VAL "1": GO SUB 2950: LET dif=VA
L "1": LET fl=PI-PI: GO TO 1620
2710 LET n=PI-PI: LET m=PI-PI: L
ET l=PI-PI
2720 LET b$="abcdef"
2730 LET c$="123456"
2740 LET m$="X": LET n$=""
2750 DIM d$(VAL "6",VAL "6"): LE
T aggr=PI-PI: LET dif=PI-PI
2760 LET a3=PI-PI: LET a4=PI-PI:
 LET fl=VAL "1"
2770 LET tot=PI-PI: LET manche=P
I-PI
2780 RETURN
2810 LET manche=manche+VAL "1":
LET tot=PI-PI
2815 IF manche=1 AND sp=1 THEN L
ET dif=(-1 AND dif=1)+(1 AND dif
=-1): LET fl=1
2820 IF manche=1 THEN LET avv=ag
gr: DIM d$(6,6): BEEP 2,30: CLS
: GO SUB 2910: GO SUB 2950: GO S
UB 3010: LET tot=-1: PRINT AT 17
,0;"##############": LET primo=(
1 AND (dif=1 OR sp=2)): GO TO 15
50
2830 IF avv<aggr THEN BEEP 1,-10
: PRINT FLASH 1;AT 19,10;"VINCE
";m$;AT 20,7;"differenza ";aggr-
avv: GO TO 2860
2840 IF avv>aggr THEN BEEP 1,-10
: PRINT FLASH 1;AT 19,10;"VINCE
";("X" AND m$="O")+("O" AND m$="
X");AT 20,7;"differenza ";avv-ag
gr: GO TO 2860
2850 BEEP VAL "1",PI-PI: PRINT F
LASH VAL "1";AT VAL "19",VAL "10
";"PARITA'!"
2860 PRINT AT 9,0;"      ";#VAL
 "1";"PER CONTINUARE PREMI UN TA
STO"
2870 PAUSE PI-PI: CLEAR : GO TO
VAL "40"
2910 PRINT INVERSE VAL "1";AT 1,
9;"ZERO E CROCI"; INVERSE PI-PI;
AT 1,25;"© 1984";AT 2,23;"A.Furl
an"
2920 PRINT AT VAL "4",PI-PI;"###
###########"
2930 RETURN
2950 PRINT AT 7,3;("ANDATA" AND
manche=0)+("RITORNO" AND manche=
1);AT 12,0;"PUNTEGGIO:": IF manc
he=1 THEN PRINT AT 14,0;"PUNTEGG
IO"'"ANDATA:"; FLASH 1;AT 15,7;a
ggr
2960 RETURN
3010 PRINT AT 4,16;"▛";AT 16,16;
"▙";AT 4,28;"▜";AT 16,28;"▟"
3020 FOR n=17 TO 27: PRINT AT 4,
n;"▀";AT 16,n;"▄": NEXT n
3030 FOR n=5 TO 15: PRINT AT n,1
6;"▌";AT n,28;"▐": NEXT n
3040 LET m=VAL "124"
3050 FOR n=0 TO 64 STEP 16: PLOT
 131,m-n: DRAW 97,0: NEXT n: IF
m=123 THEN GO TO 3070
3060 LET m=VAL "123": GO TO VAL
"3050"
3070 LET m=VAL "147"
3080 FOR n=0 TO 64 STEP 16: PLOT
 m+n,43: DRAW 0,97: NEXT n: IF m
=148 THEN GO TO VAL "3100"
3090 LET m=VAL "148": GO TO VAL
"3080"
3100 PLOT 132,44: DRAW 95,0: DRA
W 0,95: DRAW -95,0: DRAW 0,-95
3110 FOR n=5 TO 15 STEP 2: PRINT
 AT n,29;n/2-1.5: NEXT n
3120 PRINT AT 17,17;"A B C D E F
"

3130 RETURN
```



Zero e Croci con lo Spectrum

to del Difensore, abbastanza alto se si impedisce (almeno in quel verso) un filetto dell'Aggressore, piuttosto basso (sempre negativo) se viene impedito un filetto del Difensore. Tutte queste varie possibilità vengono interpretate correttamente dall'unica linea 360, grazie alla "variabilità" delle variabili "dif" e "giro" (e soprattutto al fatto che "meno per meno fa più"!).

Nelle subroutine 400 e 500 si assegna il punteggio (ridotto ad un decimo ed a un centesimo dei precedenti) in base alle minacce di filetto (o di interruzione di filetto) che si creano occupando certe caselle. Infatti quando l'Aggressore ha disposto su una linea due sue pedine accostate e con ai due lati una casella vuota, alla sua mossa successiva sicuramente formerà filetto di tre pedine (il lettore provi). Pertanto se in h$ vi è una sola pedina in seconda (o terza) posizione, l'occupazione della terza (o seconda) posizione permetterà alla successiva propria mossa di guadagnare un punto (o di impedire ciò, se a muovere è stato il Difensore). Più precisamente a questa casella verrà assegnato (routine 400) un punteggio scelto in base alle quattro eventualità già accennate (routine 300).

L'altra routine (500) si attiva invece quando la pedina si trova ad una estremità di h$: il punteggio verrà quindi assegnato alle caselle centrali di h$.

Infine la routine 600 completa la ricerca di caselle che posssono formare filetto, analizzando le quattro strisce di tre caselle ciascuna (agli angoli) sfuggite alla ricerca fatta tra la 110 e la 290.

Siccome ogni casella è interessata a più direzioni (da un minimo di tre per quelle angolari ad un massimo di otto per quelle centrali), il lettore avrà già intuito che in una stessa casella si possono verificare anche tutte le quattro possibilità viste alle routine 300 e/o 400 e/o 500. Pertanto le assegnazioni di punteggio saranno sempre aggiuntive (algebricamente).

A questo punto ad ogni casella della scacchiera corrisponde un punteggio, Spectrum deve quindi scoprire quella a punteggio maggiore per piazzarvi la propria pedina. A tale compito provvedono le linee comprese fra la 800 e la 900. (Il lettore interessato potrebbe inserire questa routine nei programmi di gioco su scacchiera che intende progettare). E' una routine completa perchè prevede anche il caso di più punteggi massimi uguali (estrazione a sorte della posizione di uno di questi).

Spectrum contro Spectrum

Questa opzione può servire inizialmente per capire a fondo le regole del gioco e in seguito per migliorare la propria strategia. La sua realizzazione è stata quanto mai semplice perchè si tratta di ridare la mossa allo stesso Spectrum dopo aver scambiato le pedine e i ruoli di Aggressore e Difensore.

Scritte in movimento entro una finestra

Per concludere, un cenno alla routine che visualizza le istruzioni di gioco; qualche lettore potrebbe volerla inserire in propri programmi, visto che è semplice e occupa poco spazio. In sostanza si tratta di muovere un testo verso l'alto entro una finestra di quattro righe sostituendo quindi ad ogni riga quella successiva. A far la parte del leone è il quasi sempre dimenticato comando Restore, utilizzato con un indirizzo variabile. Ricordo ai lettori più distratti che Restore viene usato con Read quando si vuole iniziare l'escursione ai Data da uno qualunque di questi anzichè dal primo in elenco nel programma. Inizialmente il Restore viene indirizzato al primo dei Data (la prima riga di istruzioni) che verrà stampata con le successive tre (Read i$, linee 2590-2600).

Appena è stampata la quarta riga viene incrementato di una linea il valore del puntatore j del Restore e la gestione passa ancora a Read per la stampa nella finestra della seconda, terza, quarta, quinta riga di istruzioni (che copriranno le righe stampate precedentemente). Tale semplice meccanismo si ripete fino alla stampa dell'ultima riga.

Ed ora che le spiegazioni sono terminate auguriamo buon lavoro nella digitazione del programma, e buon divertimento nell'apprendere le strategie di gioco. ■

COMMODORE VIC 20 - C 64

Appunti di utilità e Gosub con tasti funzione

Alessandro Guida

Questo mese ospitiamo i contributi di due lettori: Marco Giovenale di Castel Gandolfo (Roma) ci propone un archivio di appunti per VIC 20.
Andrea Caporin di Treviso, invece, una routine in linguaggio macchina, che permette di effettuare dei salti a subroutine con la pressione dei tasti funzione del C 64.
Grazie e complimenti a entrambi!

Appunti di utilità

Mi sono trovato spesso, programmando sul VIC, a cercare, scartabellando libri e riviste, il byte delle coordinate schermo, oppure quello del buffer della tastiera, o altri indirizzi utili.
Per ovviare a questo inconveniente ho steso due righe di appunti e li ho appesi al muro. Passati alcuni giorni ho notato che erano incompleti: mancavano parecchie notiziole apprese col tempo sul mio Commodore e altre curiosità imparate in seguito, che non avevo segnato.
Allora mi sono deciso a caricare un word processor e a fare un lavoretto preciso. Il risultato è ciò che state leggendo: una breve raccolta di notizie e piccole utility, che potrà essere consultata, anche se si presenta molto in disordine, da chi voglia aiutare i propri programmi BASIC a fare bella figura di fronte alle enormi "cabale" del linguaggio macchina.
Non ho (volutamente) inserito ciò che riguarda i caratteri e il suono, in quanto entrambi gli argomenti sono abbondantemente trattati nel manuale, nella "Guida di riferimento" e nel libro di R. Bonelli: "Alla scoperta del VIC 20"...
Spero così di fare un gradito omaggio a quanti, come me, impazziscono per colpa di uno stupido byte fuori posto o passano pomeriggi interi alla ricerca di un vecchio articolo sulla protezione dei programmi.
Con Poke 650, 128 si attiva la continuità dei tasti e Poke 650, 0 la disattiva. Con Sys 64820 il computer si resetta.
Poke 4098,16 riporta in memoria la prima riga di programma cancellata con un New; è bene ricordare, però, che questa riga può essere solo listata e mai, o quasi, eseguita senza rischiare di dover spegnere il VIC.
Dopo aver fatto girare questo programma:

```
10 PRINT PEEK (653)
20 RUN
```

avrete 1 premendo Shift, 2 premendo Commodore, 3 premendo i due tasti contemporaneamente e 4 con il Ctrl. Da ciò viene che:
Wait 653,1 attende la pressione di SH.
Wait 653,2 attende quella di CM.
Wait 653,3 quella di SH. +CM.
e Wait 653, 4 quella di CT.
Un altro trucco è usare Wait per ttendere che sia premuto un tasto; si può scrivere cioè:
Poke 198,0:Wait198,1 oppure:
Poke198,0:Wait197,X, dove X è il codice del tasto da premere.
Il numero dei caratteri presenti nel buffer della tastiera è contenuto in 198. Il buffer può memorizzare, al massimo, 10 caratteri e occupa i byte da 631 e 640.
I byte 197 e 203 danno il valore del tasto premuto in uno speciale codice Commodore. 215 invece restituisce il valore in ASCII.
La locazione 36864 di solito contiene 12, nella 36865 invece ci dovrebbe essere un 38. Se si alterano questi valori si sposta lo schermo...:

```
10 POKE 36864,5:POKE36865,100
15 FORT = 1TO1000:NEXT
20 POKE36864,12
25 FORT = 120 TO 38 STEP -1
30 POKE 36865, T
35 NEXT
```

Così si cambiano le coordinate dello schermo.
Provate poi a "pokare" nella locazione 36869 dei valori a caso, diverse volte, vedrete che succede! È il byte che controlla la mappa di caratteri del VIC...
Ecco poi un'utile routine per far apparire lentamente le lettere:

```
10 PRINT CHR$ (147):?"<2 DOWN> <RIGHT>";
15 READ T$ :POKE36878,15
20 IF T$ "BASTA" THEN POKE36878,0:END
30 FORI = 1 TO LEN (T$)
40 PRINT MID$ (T$, I, 1); :POKE36877,200
50 FORT = 1TO50:NEXTT:POKE36877,0:NEXTI:?
60 PRINT "<DOWN> <RIGHT>";
70 GOTO 15
100 DATA "DOWN È UN CURSORE IN"
110 DATA "BASSO.RIGHT VERSO DE-"
120 DATA "STRA."
130 DATA "-------------------------------------------"
140 DATA "BASTA"
```



Appunti di utilità e Gosub con tasti funzione

Listato 1. *Sorgente commentato della routine.*

```
ONKEYSRC......PAGE 0001

LINE# LOC   CODE         LINE

00001  0000                 ;          ON KEY GOSUB
00002  0000                 ;
00003  0000                 ;      BY ANDREA CAPORIN
00004  0000                 ;
00005  0000                 *=$C000
00006  C000  A9 30                LDA #<ONKEY     ;MODIFICA IL VETTORE
00007  C002  8D 28 03             STA $0328       ;DELLA ROUTINE DI STOP
00008  C005  A9 C0                LDA #>ONKEY
00009  C007  8D 29 03             STA $0329
00010  C00A  A9 00                LDA #$00        ;AZZERA LA LOCAZIONE
00011  C00C  85 FB                STA $FB         ;DEL NUMERO DI TASTI ATTIVATI
00012  C00E  A8                   TAY
00013  C00F  20 79 00     RIT     JSR $0079       ;PRELEVA L'ATTUALE BYTE DEL TESTO
00014  C012  B0 12                BCS NODEC       ;SE NON E' DECIMALE SALTA
00015  C014  E6 FB        SAL     INC $FB         ;INCREMENTA DI UNO IL NUMERO DI TASTI ATTIVATI
00016  C016  20 6B A9             JSR $A96B       ;LEGGE UN INTERO
00017  C019  A5 14                LDA $14         ;DAL TESTO
00018  C01B  99 34 03             STA $0334,Y     ;E LO DEPOSITA
00019  C01E  A5 15                LDA $15
00020  C020  99 38 03             STA $0338,Y
00021  C023  C8                   INY
00022  C024  D0 E9                BNE RIT
00023  C026  C9 2C        NODEC   CMP #$2C        ;SE IL CARATTERE SEGUENTE NON E'
00024  C028  D0 05                BNE FINE        ;UNA VIRGOLA TORNA AL BASIC
00025  C02A  20 73 00             JSR $0073       ;ALTRIMENTI LEGGE UN NUOVO CARATTERE
00026  C02D  90 E5                BCC SAL         ;SE DECIMALE RIPETE
00027  C02F  60           FINE    RTS
00028  C030  A5 C5        ONKEY   LDA $C5
00029  C032  C9 40                CMP #$40        ;SE E' PREMUTO UN TASTO
00030  C034  D0 05                BNE CONT        ;VA AL SUCCESSIVO TEST
00031  C036  85 FC                STA $FC         ;ALTRIMENTI SALTA
00032  C038  4C ED F6     STOP    JMP $F6ED       ;ALLA NORMALE ROUTINE DI STOP
00033  C03B  AE 8D 02     CONT    LDX $028D       ;SE NON E' PREMUTO
00034  C03E  F0 F8                BEQ STOP        ;CONTEMP. LO SHIFT OPPURE
00035  C040  C5 FC                CMP $FC         ;E' LO STESSO TASTO DEL PRECEDENTE
00036  C042  F0 F4                BEQ STOP        ;TEST TERMINA
00037  C044  85 FC                STA $FC
00038  C046  A6 FB                LDX $FB         ;NUMERO DI TASTI FUNZ. ATTIVATI
00039  C048  DD 75 C0     LOOP    CMP COD-1,X     ;SE E' UN TASTO
00040  C04B  F0 05                BEQ ENDL        ;FUNZ. ATTIVATO PROSEGUE
00041  C04D  CA                   DEX
00042  C04E  D0 E8                BNE LOOP
00043  C050  F0 E6                BEQ STOP        ;ALTRIMENTI TERMINA
00044  C052  BD 33 03     ENDL    LDA $0334-1,X   ;PRELEVA IL NUMERO
00045  C055  85 14                STA $14         ;DI LINEA RELATIVO
00046  C057  BD 37 03             LDA $0338-1,X   ;AL TASTO PREMUTO E
00047  C05A  85 15                STA $15         ;LO IMMETTE IN $14 E $15
00048  C05C  A9 03                LDA #$03        ;PARTE DELLA ROUTINE
00049  C05E  20 FB A3             JSR $A3FB       ;GOSUB DELL'INTERPRETE
00050  C061  A5 7B                LDA $7B         ;BASIC CHE
00051  C063  48                   PHA             ;MEMORIZZA NELLO STACK
00052  C064  A5 7A                LDA $7A         ;L'ATTUALE CARATTERE
00053  C066  48                   PHA             ;DEL TESTO BASIC
00054  C067  A5 3A                LDA $3A         ;IL CORRENTE
00055  C069  48                   PHA             ;NUMERO DI RIGA
00056  C06A  A5 39                LDA $39
00057  C06C  48                   PHA
00058  C06D  A9 8D                LDA #$8D        ;E IL CODICE TOKEN
00059  C06F  48                   PHA             ;DELL'ISTRUZIONE GOSUB
00060  C070  20 A3 A8             JSR $A8A3       ;QUINDI SALTA ALLA ROUTINE GOTO
00061  C073  4C AE A7             JMP $A7AE       ;ED ESEGUE LA NUOVA LINEA
00062  C076  04           COD     .BYT 4,5,6,3    ;CODICE DEI QUATTRO T. FUNZ.
00062  C077  05
00062  C078  06
00062  C079  03
00063  C07A                      .END
ERRORS = 00000
SYMBOL TABLE
SYMBOL VALUE
 COD      C076    CONT     C03B    ENDL     C052    FINE     C02F
 LOOP     C048    NODEC    C026    ONKEY    C030    RIT      C00F
 SAL      C014    STOP     C038
```



Appunti di utilità e Gosub con tasti funzione

Si possono mettere quante Data si vogliono, l'importante è ricordare, quando non ne servono altre, di battere:

DATA "BASTA"

come nella riga 140.
Vi rammento come cambiare l'altezza dei caratteri:

PRINT CHR$ (14)...maiuscoli e minuscoli;
PRINT CHR$ (142)..solo maiuscoli;
PRINT CHR$ (8)....per bloccarli;
PRINT CHR$ (9)....per riattivarli.

Con Poke 808,100 potrete proteggere un programma da qualsiasi tentativo di blocco. Attenzione, però, il listato sarà illeggibile anche per voi, a meno che ristabiliate il valore consueto con Poke 808,...inserite voi il numero.
Per un'ulteriore protezione potete digitare Poke 788,194 (blocca il tasto Stop) e/o Poke 37150,3 (blocca il tasto Restore).
Per far impazzire il computer si procede nel seguente modo, almeno nella maggior parte dei casi.
1- Si spegne il VIC e lo si riaccende.
2- Si digita Poke 4098,16.
3- Si dà il List...già è iniziato lo squilibrio mentale.
4- Si dà il Run e si ottiene così un messaggio di errore.
5- Si cancella la linea che il VIC ci indica come sbagliata.

Esempio:
Voi digitate Run
e il VIC risponde con:

?SYNTAX
ERROR IN 56.

Allora battete 56 e premete Return...
Il VIC darà chiari segni di instabilità psicofisica.
6- Spegnere il computer è l'unico modo per riportare tutto alla normalità.
A volte questo procedimento non funziona: dipende dall'umore del VIC. Si può allora adottare il Sistema Giovenale, adatto a far impazzire *qualunque* computer funzionante in BASIC (credo), con le dovute modifiche. Il Metodo G. consiste nel battere questo programma:

```
10 POKE36879,110
20 PRINT CHR$ (147); CHR$ (142)
30 PRINT CHR$ (8); CHR$ (5)
40 L = 255 ★ RND (1)
50 D = INT (65535 ★ RND (1))
60 J = INT (65535 ★ RND (1))
70 POKE D,L
80 PRINT "POKE" D "," L:PRINT
90 POKE J,0
100 PRINT "POKE" J "," 0:PRINT
110 GOTO40
```

Spesso occorre molto tempo per un "*impazzimento totale*", ma si può essere certi che arriverà!
Non volendo perder tempo, si può scrivere:

```
10 FORI = 0TO255
20 POKEI,255 ★ RND (1)
30 NEXT
40 RUN
```

Oppure:

```
5 FORT = 1TO500
10 PRINTCHR$ (5);C = INT (RND(1) ★ 94) + 33
20 PRINTCHR$ (C); :NEXT
30 POKE 36879,8
40 POKE 36869,255
50 POKE 36878,15
60 FORT = 12 TO 60
70 POKE 36877,RND (1) ★ 255
80 POKE 36874, RND (1) ★ 255
90 POKE 36864,T:NEXT
100 POKE808,100:POKE773,100
110 FORT = 0TO255:POKE T,0:NEXT
120 RUN30
```

Un'istruzione *anti-tutto* è la seguente:
POKE 773,100

oppure:
POKE 773,198

oppure:
POKE 773,196

oppure:
POKE 773,255

Listato 2. *Il programma caricatore BASIC della routine.*

```
10 REM*    ESEMPIO DIMOSTRATIVO    *
20 REM*DELL'ISTRUZIONE ON KEY GOSUB*
30 SYS49152,50,70
40 GOTO40
50 PRINTCHR$(147)"IL SALTO A QUESTA LINEA E' ABILITATO"
60 PRINT"DAL TASTO"CHR$(18)" 'F2' (SHIFT+F1)":RETURN
70 PRINTCHR$(145)"DAL TASTO"CHR$(18)" 'F4' (SHIFT+F3)":RETURN
READY.
```

Listato 3. *Un esempio dimostrativo.*

```
5 REM CARICATORE BASIC
10 PRINTCHR$(147)"ON KEY GOSUB"
20 FORI=49152TO49273:READA:X=X+A:POKEI,A:NEXT
30 IFX<>15440THENPRINT"ERRORE NEI DATA !":END
40 DATA169,48,141,40,3,169,192,141,41,3,169,0,133,251,168,32,121,0,176,18,230
50 DATA251,32,107,169,165,20,153,52,3,165,21,153,56,3,200,208,233,201,44,208,5
60 DATA32,115,0,144,229,96,165,197,201,64,208,5,133,252,76,237,246,174,141,2,240
70 DATA248,197,252,240,244,133,252,166,251,221,117,192,240,5,202,208,248,240,230
80 DATA189,51,3,133,20,189,55,3,133,21,169,3,32,251,163,165,123,72,165,122,72
90 DATA165,58,72,165,57,72,169,141,72,32,163,168,76,174,167,4,5,6,3
READY.
```

Digitando Poke 37148,X si blocca il registratore, se X è minore di 252.
La locazione 36866, cambiata, fa diminuire il numero delle colonne dello schermo; la 36867 controlla le righe.
Come è anche spiegato nel manuale, ogni istruzione del BASIC si ottiene, volendo abbreviarla, con la prima lettera + la seconda shiftata, tranne alcune eccezioni. Per esempio Load = L + (Shift + O).
Le eccezioni sono:

- CLOSE..............CL + (SHIFT + O).
- GOSUB.............GO + (SHIFT + S).
- INPUT...............inesistente.
- PRINT...............?.
- RESTORE.........RE + (SHIFT + S).
- RETURN..........RE + (SHIFT + T).
- STEP..................ST + (SHIFT + E).
- LEFT$...............LE + (SHIFT + F).
- STR$..................ST + (SHIFT + R).

Volendo simulare l'istruzione Print At X,Y sul ristretto BASIC del VIC, si può procedere come segue (**Personal Software**, Luglio-Agosto '83, pag. 93):

POKE 211,X: POKE 214,Y: SYS 58759

L'indirizzo 204 consente il controllo del lampeggio cursore.
Se Peek (204) uguale 0, il cursore lampeggia; se Peek (204) uguale 1, il cursore non appare. Un esempio può essere:

```
10 GETA$:IFA$ = " " THENPOKE204,0:GOTO10
20 PRINTA$;
30 A = A + 1 : IFA = 5 THEN PRINT:PRINT:A = 0
40 GOTO10
```

Gosub con tasti funzione

Tra le numerosissime istruzioni presenti negli interpreti BASIC dell'ultima generazione spicca senz'altro la possibilità di saltare a una determinata linea di programma al verificarsi di certi eventi, tra i quali la pressione di un tasto funzione.
Normalmente il salto viene effettuato sfruttando l'interrupt del microprocessore, cioè sessanta volte al secondo (nel caso del C 64) la routine di interrupt controlla se è premuto un tasto, in precedenza attribuito a un certo numero di linea. In caso affermativo si salva l'indirizzo di ritorno e l'esecuzione prosegue dal nuovo numero di linea, per poi, alla presenza di un Return, tornare al punto in cui il programma era stato interrotto. Proprio come una normale chiamata di subroutine.
Questa possibilità ovviamente non è presente nello scarno BASIC del C 64, però si può simulare mediante una breve routine in linguaggio macchina.

Per prima cosa, da come si può osservare dal codice sorgente della routine, non ho utilizzato la possibilità di modificare l'IRQ, che dava dei problemi dovuti essenzialmente alla brusca interruzione del programma BASIC in esecuzione, ma sfruttando piuttosto la chiamata della routine di Stop, la quale controlla se questo tasto è stato premuto. Questa chiamata viene effettuata prima dell'esecuzione di ogni istruzione BASIC.

Il listato 1 sorgente è completamente commentato. La routine nella parte iniziale, dopo il vettore di Stop, cerca i numeri di linea separati da una virgola, a ognuno di questi assegna un tasto funzione, partendo dal primo in alto assieme allo Shift, quindi torna al BASIC.
Da questo momento i tasti funzione interessati sono attivati per tutta l'esecuzione del programma, ad eccezione di quando il programma è in attesa di un Input, in quanto non viene sentito il tasto di Stop.
La sintassi del comando è la seguente:

SYS49152,n1,n2,n3,n4

dove n1...n4 sono i numeri di linea da attribuire ai tasti funzione, a partire dal primo in alto, doppiati al tasto di Shift. I salti che si possono effettuare vanno da un minimo di 1 a un massimo di 4.
La routine può essere facilmente rilocata in altre zone di memoria, cambiando i valori contenuti in $C001-$C006, e $C049-$C04A.
Per utilizzare il programma occorre ricopiare il listato 2 e lanciarlo in esecuzione: da questo punto in poi è possibile richiamare Sys 49152 secondo la sintassi precedente. A titolo di esempio di utilizzo è riportato il programma dimostrativo del listato 3.

SINCLAIR ZX SPECTRUM

Come (tentare di) recuperare programmi BASIC

Marcello Spero

Questo mese presentiamo un interessante contributo di Luciano Andrei, abitante a Lastra a Signa (FI) - I due programmi proposti, Lazzaro 1 e Lazzaro 2, permettono: il primo di recuperare i programmi in BASIC illeggibili da un certo punto in poi (senza coda) e il secondo di salvare quelli senza header. Entrambi i programmi sono scritti in linguaggio macchina.

Lazzaro 1

Tutti sanno che se sul nastro c'è un errore, il Load si interrompe e dando il List da un certo punto in poi c'è solo caos: appare cioè una linea senza senso, mentre



Come (tentare di) recuperare programmi BASIC

prima tutto è normale. Provando a cancellare tale linea, numero di linea ed Enter, per poter utilizzare la parte corretta, il più delle volte cade il sistema e se va bene il problema sorge per la linea successiva. A volte capita di perdere una sola linea e di dover ridigitare tutto il programma, con notevole perdita di tempo. Il mio programma risolve il problema, cancellando tutto quello che non va. Il nome è dovuto proprio a questo, infatti "lazzarizza" i programmi, cioè li resuscita. Il programma (figura 1) va caricato nel buffer della stampante con Load "Lazzaro 1" Code 23296, poi si carica il programma sciupato ed appena appare il segnale di errore si ferma il registratore, si dà il List ed allo "scroll"? si

Figura 1. *I codici del programma Lazzaro 1: vanno caricati e memorizzati con un semplice programma caricatore BASIC, del tipo di quelli più volte pubblicati in questa rubrica.*

23296	42	23344	35
23297	93	23345	54
23298	92	23346	58
23299	62	23347	35
23300	58	23348	54
23301	190	23349	234
23302	32	23350	201
23303	10	23351	35
23304	62	23352	35
23305	14	23353	35
23306	35	23354	78
23307	190	23355	35
23308	40	23356	70
23309	41	23357	42
23310	61	23358	83
23311	190	23359	92
23312	32	23360	86
23313	246	23361	35
23314	62	23362	94
23315	2	23363	35
23316	205	23364	235
23317	1	23365	175
23318	22	23366	237
23319	6	23367	66
23320	6	23368	229
23321	33	23369	235
23322	31	23370	94
23323	21	23371	35
23324	126	23372	86
23325	230	23373	35
23326	127	23374	25
23327	35	23375	209
23328	215	23376	122
23329	16	23377	179
23330	249	23378	32
23331	42	23379	236
23332	93	23380	34
23333	92	23381	75
23334	35	23382	92
23335	54	23383	24
23336	234	23384	209
23337	201	23385	0
23338	42	23386	0
23339	93	23387	0
23340	92	23388	0
23341	35	23389	0
23342	54	23390	0
		23343	253

Figura 2. *Assembly di Lazzaro 1.*

```
10 ;   LAZZARO 1
20
30 ;   ANDREI LUCIANO
40
50
60 CHADD          EQU 23645
70 VAR            EQU 23627
80 PROG           EQU 23635
90
100               LD HL,(CHADD)
110               LD A,":"
120               CP (HL)
130               JR NZ,ERRORE
140 A1            LD A,14
150               INC HL
160               CP (HL)
170               JR Z,A3
180               DEC A
190               CP (HL)
200               JR NZ,A1
210 ERRORE        LD A,2
220               CALL #1601
230               LD B,6
240               LD HL,#151F
250 A2            LD A,(HL)
260               AND 127
270               INC HL
280               RST 16
290               DJNZ A2
300               LD HL,(CHADD)
310               INC HL
320               LD (HL),#EA ;REM
330               RET
340 A4            LD HL,(CHADD)
350               INC HL
360               LD (HL),#FD ;CLEAR
370               INC HL
380               LD (HL),":"
390               INC HL
400               LD (HL),#EA ;REM
410               RET
```

risponde con Enter fino a quando appare la linea sciupata, dopodiché si risponde con Break. A questo punto si digita, senza numero di linea, Randomize Usr 23296 : Print nn dove nn è il numero dell'ultima linea corretta, ed il gioco è fatto. Adesso il programma può essere completato e tutto è a posto.

Il programma è rilocabile e può anche servire per eliminare un programma in BASIC da una certa linea in poi. Vediamone adesso il funzionamento: il programma cerca l'indirizzo dopo l'ultima linea corretta e lo pone in Vars (Vars indica al sistema operativo dove inizia la zona delle variabili), dopo di che modifica la linea che lo ha chiamato (Randomize Usr 23296 : Print nn) in Randomize Usr 23296 : Clear : Rem..., in modo che la successiva istruzione che il computer trova è un Clear, per cui cancella tutte le variabili ed il nonsenso che ci abbiamo messo dentro. Un modo di vedere il cambiamento è il seguente programmino:

```
10 RANDOMIZE USR 23296 : PRINT 20
20 REM
30 REM e poi RUN.
```



Come (tentare di) recuperare programmi BASIC

Seguito listato Assembly Lazzaro 1.

```
A3      INC HL
        INC HL
        INC HL
        LD C,(HL)
        INC HL
        LD B,(HL)
        LD HL,(PROG)
A5      LD D,(HL)
        INC HL
        LD E,(HL)
        INC HL
        EX DE,HL
        XOR A
        SBC HL,BC
        PUSH HL
        EX DE,HL
        LD E,(HL)
        INC HL
        LD D,(HL)
        INC HL
        ADD HL,DE
        POP DE
        LD A,D
        OR E
        JR NZ,A5
        LD (VAR),HL
        JR A4

;       END
```

Lazzaro 2

Se si sciupa l'header di un programma in BASIC, è praticamente impossibile caricarlo e farlo funzionare o ricostruirci l'header.
Lazzaro 2 invece può permetterci di farne una copia corretta con l'header. Il programma (figura 2), senza header viene caricato come un Code, ma messo nella zona riservata al BASIC (da Prog in poi) e poi, supposto che sia in BASIC, aggiusta i puntatori del sistema operativo (Vars inizio variabili ed Eline fine variabili), distrugge le variabili (vedremo dopo come evitarlo) e restituisce il controllo al sistema: a questo punto basta fare un Save... ed il gioco è fatto. Il programma va caricato nel buffer della stampante (Load "" Code 23296). Se il nastro è rumoroso, o di seguito c'è registrato un altro programma, il Load ottenuto con Lazzaro 2 continua, ed è bene spegnere il registratore appena si riconosce il fischio di inizio blocco. Se il nastro è nuovo, non ci sono problemi, comunque, prima di effettuare il Save... è bene dare un Clear e poi con un List controllare la presenza di tutte le linee. Se però la parte più importante non è il programma BASIC, ma le variabili, oppure si vuole resuscitare un blocco Data, bisogna provare come segue, anche se il risultato non sempre è assicurato.
Fare Poke 23378,0 : Poke 23379,0 e poi Enter, regola-



Come (tentare di) recuperare programmi BASIC

re il volume per tentativi, in modo che il Load si fermi alla fine del programma, e poi attivare Lazzaro 2 normalmente, cioè Randomize Usr 23296 seguito da Enter e far partire il registratore. Senza le variabili il funzionamento è assicurato, se poi si vogliono anche quelle, a volte va male!.

Veniamo adesso al funzionamento. La struttura di una linea in BASIC è mostrata nel 24° capitolo pagina 122 del manuale originale; come si vede, ci sono due byte per il numero di linea, due per la lunghezza della linea più 1, il corpo della linea e per finire un Enter (13). Supposto allora che il programma sia in BASIC, il terzo e quarto byte, sommati al valore attuale (inizio corpo linea), danno un indirizzo che deve contenere 13, altrimenti siamo fuori della zona BASIC e si termina. Se invece c'è il numero 13, ci sono buone probabilità di essere su di

Figura 3. *I codici di Lazzaro 2.*

23296	193	23344	6
23297	49	23345	35
23298	254	23346	126
23299	91	23347	43
23300	197	23348	187
23301	62	23349	56
23302	255	23350	18
23303	221	23351	221
23304	42	23352	229
23305	83	23353	209
23306	92	23354	124
23307	33	23355	186
23308	64	23356	40
23309	156	23357	4
23310	55	23358	56
23311	205	23359	217
23312	86	23360	24
23313	5	23361	6
23314	221	23362	125
23315	34	23363	187
23316	89	23364	56
23317	92	23365	211
23318	42	23366	40
23319	83	23367	7
23320	92	23368	43
23321	86	23369	175
23322	35	23370	237
23323	94	23371	66
23324	35	23372	43
23325	78	23373	43
23326	35	23374	43
23327	70	23375	34
23328	9	23376	75
23329	126	23377	92
23330	254	23378	54
23331	13	23379	128
23332	32	23380	207
23333	35	23381	255
23334	35	23382	92
23335	126	23383	24
23336	254	23384	209
23337	40	23385	0
23338	48	23386	0
23339	35	23387	0
23340	186	23388	0
23341	56	23389	0
23342	32	23390	0
23343	32		

Figura 4. *Assembly di Lazzaro 2.*

```
10
20 ; LAZZARO 2
30
40 ; ANDREI LUCIANO
50
60
70 PROG          EQU 23635
80 ELINE         EQU 23641
90 VAR           EQU 23627
100
110              ENT $
120
130              POP BC
140              LD SP,23550
150              PUSH BC
160              LD A,255
170              LD IX,(PROG)
180              LD HL,40000
190              SCF
200              CALL 1366
210              LD (ELINE),IX
220
230              LD HL,(PROG)
240 ANCORA       LD D,(HL)
250              INC HL
260              LD E,(HL)
270              INC HL
280              LD C,(HL)
290              INC HL
300              LD B,(HL)
310              ADD HL,BC
320              LD A,(HL)
330              CP 13
340              JR NZ,FINE0
350              INC HL
360              LD A,(HL)
370              CP 40
380              JR NC,FINE1
390              CP D
400              JR C,FINE1
410              JR NZ,POI
420              INC HL
430              LD A,(HL)
440              DEC HL
450              CP E
460              JR C,FINE0
470 POI          PUSH IX
480              POP DE
490              LD A,H
500              CP D
510              JR Z,A1
520              JR C,ANCORA
530              JR FINEA
540 A1           LD A,L
550              CP E
560              JR C,ANCORA
570              JR Z,FINE1
580 FINEA        DEC HL
590 FINE0        XOR A
600              SBC HL,BC
610              DEC HL
620              DEC HL
630              DEC HL
```



Seguito figura 4.

```
2372 (VAR),HL
680               JR A4
690
700 ;        END
0
640 FINE1         LD (VAR),HL
650               LD (HL),128
660 FINE2         RST 8
670               DEFB 255
680
690
700 ;        END
```

una linea BASIC, si controlla allora se il numero di questa ipotetica linea è maggiore di quello della precedente, se non lo è: allora come ultima linea valida per il BASIC viene presa la precedente. Altrimenti il gioco prosegue fino a che si verifica un "errore" dei precedenti o siamo oltre la zona caricata da nastro.
Alla fine vengono aggiustati i puntatori e restituito il controllo.

Siccome Lazzaro 2 sposta lo Stack Pointer nel buffer della stampante, è consigliabile, prima di passare ad un altro tipo di lavoro, dare Randomize Usr 0 per resettare completamente il computer, perché a volte si possono verificare crash del sistema operativo.

SHARP MZ-700

Struttura interna delle istruzioni BASIC

Martino Sangiorgio

Vi siete mai chiesti come lo Sharp MZ-700 compatti in memoria le istruzioni BASIC? No? Eppure può essere non solo interessante, ma anche utile, come vedremo più avanti. L'analisi che condurremo risulta, nell' MZ-700, semplificata dalla presenza di un programma Monitor di sistema, ma soprattutto dalla presenza, nello stesso interprete BASIC, di un programma Monitor ancora più sofisticato del primo. Il programma BASIC che analizzeremo è quello visibile nella figura 1.
Carichiamo quindi il BASIC da cassetta e iniziamo a digitare la prima istruzione:

10 PRINT "C"

Per vedere ora come tale istruzione è stata registrata in memoria dobbiamo passare il controllo al Monitor del BASIC. Per fare ciò digitiamo il comando Bye e premiamo il tasto Cr. Ora che siamo in ambiente Monitor possiamo digitare il comando:

D6BC0

che ci permette di visualizzare il contenuto di un blocco di memoria, partendo dall'indirizzo 6BC0h. L'indirizzo di inizio del programma BASIC (trovato dopo vari tentativi) è 6BCFh. Quello che si nota da questo indirizzo in poi è visibile nella figura 2. In essa si notano due valori esadecimali 0Ah, cioè 10d. Uno dei due potrebbe essere il numero di linea, ma l'altro? Tra gli altri valori numerici riconosciamo il 22h, codice ASCII del doppio apice, e il 16h, codice ASCII del carattere C. Quindi gli altri due (8Fh e 20h) dovrebbero individuare la codifica dell'istruzione Print. Ma la nostra analisi è solo all'inizio.

Ritorniamo in ambiente BASIC digitando il comando Monitor R (seguito da CR) e digitiamo l'istruzione successiva:

20 INPUT "MESSAGGIO";A

Ritorniamo di nuovo al Monitor col comando Bye e vediamo (sempre col comando D6BC0) cosa c'è attualmente in memoria.

Vedremo l'insieme dei valori esadecimali visibili in figura 3.
In essi riconosciamo quelli già visti prima. Il primo valore della seconda istruzione inizia all'indirizzo 6BD9h. Contando i byte della prima istruzione ci accorgiamo che essi sono effettivamente 10: quindi uno dei due 0Ah iniziali rappresenta la lunghezza dell'istruzione. Purtroppo, anche nella seconda istruzione abbiamo due valori uguali (14h), sia per il numero di linea che per la lunghezza dell'istruzione. Vedremo quindi successivamente di scoprire quale dei due determina la lunghezza. Nella seconda istruzione, al quinto e sesto byte, notiamo un 91h e un 20h, che sono in corrispondenza ai valori 8Fh e 20h della prima. Poiché ora l'istruzione è diversa, ne deduciamo che 8Fh dovrebbe essere la codifica della Print e 91h la codifica della Input, mentre il 20h, che è comune alle due istruzioni, deve avere un altro significato. Gli altri valori presenti nella seconda istruzione non sono altro che le codifiche ASCII di quello che abbiamo scritto dopo la Input.
Ritorniamo di nuovo al BASIC, digitando R (seguito da Cr) e scriviamo la terza istruzione:

30 IF A = 0 THEN GOTO 50

Rapidamente (ormai abbiamo capito come si fa) scriviamo Bye e, appena nel Monitor, D6BC0. Vedremo ora i valori mostrati in figura 4.

L'indirizzo di inizio della terza istruzione è il 6BEDh e, contando i byte dall'inizio della seconda istruzione (il primo 14h), notiamo che essi sono proprio 20 (infatti 14h = 20d). Ora, dall'indirizzo 6BEDh, notiamo un 17h (= 23d) e un 1Eh (= 30d). Poiché 30 è proprio il

numero di riga, l'altro sarà necessariamente la lunghezza dell'istruzione. Proseguendo, notiamo: un 93h, che dovrebbe corrispondere alla codifica della If; un 20h, che ancora non capiamo, ma che finora ha accompagnato tutte le istruzioni; un 41h, codice ASCII della variabile A; un F4h, che dovrebbe rappresentare il segno di uguale; un 15h, che non capiamo, ma che dovrebbe essere associato al numero seguente che è uno zero (ci sono cinque byte a zero). Proseguendo, notiamo: un 20h, che non capiamo; un gruppo formato da E2h e da 20h, che dovrebbe essere la codifica di Then; un gruppo formato da 80h e da 20h, che dovrebbe essere la codifica di Goto; uno 0Bh, che non capiamo, e un 32h (= 50d), che è il numero di linea dopo il Goto.

Ora il meccanismo comincia ad essere un po' più chiaro. Digitiamo R (seguito da Cr) e, quando siamo nel BASIC, scriviamo le due ultime istruzioni:

40 B = 3.1415 ★A↑ 2:PRINT "AREA=";B:GOTO 20
50 END

Di nuovo scriviamo Bye e, nel Monitor, D6BC0. La conformazione della memoria che ci interessa sarà uguale a quella visibile in figura 5.

L'istruzione 30 (la If) iniziava all'indirizzo 6BEDh col valore 17h (= 23d). Se questa è veramente la lunghezza dell'istruzione, dopo 23 byte dovremmo trovare l'inizio della successiva. Infatti, all'indirizzo 6C04h troviamo: il valore 2Ah (= 42d), lunghezza dell'istruzione con numero di linea 40; un byte a zero; un valore 28h (= 40d), numero della linea; un altro byte a zero; il valore 42h (codice ASCII della B); il valore F4h, che dovrebbe rappresentare il segno di uguale, e il valore 15h, che dovrebbe far capire che quello che segue è un valore numerico. Poiché la costante numerica utilizzata (3,1415) è un numero reale, dovrebbe occupare 4 byte di memoria. Troviamo invece una serie di cinque byte:

82 - 49 - 0E - 56 - 04

Dopo qualche calcolo sommario abbandoniamo l'idea che si tratti della codifica di un numero in virgola mobile. Deve trattarsi di una particolare codifica Sharp. Accantoniamo per il momento lo studio della costante reale e proseguiamo nella lettura della memoria. All'indirizzo seguente (6C10h) troviamo: FCh, codifica dell'operatore ★; 41h (codice ASCII della A); FDh, codifica dell'operatore ↑; 15h, che è posto prima di un valore numerico, e un gruppo di cinque byte, di cui il primo ha valore 82h e gli altri quattro sono a zero. Questo gruppo dovrebbe essere la codifica della costante 2, che viene quindi considerata dal sistema come reale, non intera.

Proseguendo, troviamo: 3Ah (codice ASCII del carattere:); 8Fh, che è la codifica dell'istruzione Print; il valore 20h, che accompagna sempre le istruzioni, e tutta una serie di valori, fino all'indirizzo 6C26h, che sono le codifiche ASCII di quello che avevamo scritto dopo Print.

Figura 1. *Listato del programma BASIC da analizzare.*

```
10 PRINT "▣"
20 INPUT "MESSAGGIO";A
30 IF A=0 THEN GOTO 50
40 B=3.1415*A↑2:PRINT "AREA = ";B:GOTO 2
0.
50 END
```

Figura 2. *Dump della zona di memoria in cui è memorizzata la prima istruzione del programma BASIC.*

```
:6BCF=0A 00 0A 00 8F 20 22 16 /....8F ".
:6BD7=22 00 00 00 00 00 00 00 /".......
```

Figura 3. *Dump della zona di memoria in cui sono memorizzate le prime due istruzioni del programma.*

```
:6BCF=0A 00 0A 00 8F 20 22 16 /....8F ".
:6BD7=22 00 14 00 14 00 91 20 /".....91
:6BDF=22 4D 45 53 53 41 47 47 /"MESSAGG
:6BE7=49 4F 22 3B 41 00 00 00 /IO";A...
```

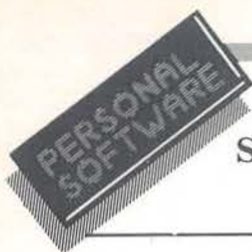
Figura 4. *Dump della zona di memoria in cui sono memorizzate le prime tre istruzioni del programma.*

```
:6BCF=0A 00 0A 00 8F 20 22 16 /....8F ".
:6BD7=22 00 14 00 14 00 91 20 /".....91
:6BDF=22 4D 45 53 53 41 47 47 /"MESSAGG
:6BE7=49 4F 22 3B 41 00 17 00 /IO";A...
:6BEF=1E 00 93 20 41 F4 15 00 /..` AF4..
:6BF7=00 00 00 00 20 E2 20 80 /.... E2 }
:6BFF=20 0B 32 00 00 00 00 00 / .2.....
```

Figura 5. *Dump della zona di memoria in cui è memorizzato l'intero programma BASIC.*

```
:6BCF=0A 00 0A 00 8F 20 22 16 /....8F ".
:6BD7=22 00 14 00 14 00 91 20 /".....91
:6BDF=22 4D 45 53 53 41 47 47 /"MESSAGG
:6BE7=49 4F 22 3B 41 00 17 00 /IO";A...
:6BEF=1E 00 93 20 41 F4 15 00 /..` AF4..
:6BF7=00 00 00 00 20 E2 20 80 /.... E2 }
:6BFF=20 0B 32 00 00 2A 00 28 / .2..*.(
:6C07=00 42 F4 15 82 49 0E 56 /.BF4.82I.
V
:6C0F=04 FC 41 FD 15 82 00 00 /.↓AFD.82.

:6C17=00 00 3A 8F 20 22 41 52 /..:8F "AR
:6C1F=45 41 20 3D 20 22 3B 42 /EA = ";B
:6C27=3A 80 20 0B 14 00 00 06 /:} .....
:6C2F=00 32 00 98 00 00 00 00 /.2.h....
```



Struttura interna delle istruzioni BASIC

All'indirizzo 6C27h ritroviamo: il valore 3Ah (codice ASCII del carattere :); 80h, codifica dell'istruzione Goto; il solito 20h; un valore 0Bh, che non comprendiamo, e infine il valore 14h (= 20d), che è il numero di linea associato al Goto. Se contiamo il numero di byte di quest'ultima istruzione, partendo dall'indirizzo 6C04h e comprendendo gli ultimi due byte a zero, vediamo che sono effettivamente 42.

All'indirizzo 6C2Eh inizia poi l'ultima istruzione: dopo uno 06h (= 6d), lunghezza dell'istruzione, un byte a zero, un valore 32h (= 50d), numero di linea dell'istruzione, e un altro byte a zero, troviamo il valore 98h, che è la codifica dell'istruzione End.

L'analisi che abbiamo fin qui condotto ci ha permesso di conoscere un po' meglio la struttura interna delle istruzioni BASIC. È pur vero che molte sono ancora le informazioni incognite, ma lo scopo che ci eravamo proposti era di dare a tutti la chiave per poter effettuare, in proprio, successive analisi più approfondite.

Abbiamo anche visto che non esiste una struttura valida per tutte le istruzioni, però qualche informazione è effettivamente comune a tutte le strutture viste. Tali informazioni possono essere sintetizzate nelle due strutture visibili in figura 6. Finora il numero di linea dell'istruzione BASIC era contenuto entro il valore 255. Per valori superiori (e fino a 65.535) servono due byte. Il valore più basso si ha nel primo dei due, il più alto nel secondo.

Provate con un numero di linea superiore a 255 e inferiore a 65.535.

Per terminare l'argomento facciamo un'ultima prova. Rientriamo sotto BASIC, digitiamo New (seguito da Cr) per cancellare il programma precedente e poi digitiamo:

10 END

Ritorniamo, col comando Bye, al Monitor e guardiamo cosa c'è dall'indirizzo 6BCFh. Vedremo:

06 - 00 - 0A - 00 - 98

(tutti valori esadecimali), che, come abbiamo visto in precedenza, è la struttura di tale istruzione.

La codifica di End, il valore 98h, si trova all'indirizzo 6BD3h. Rientriamo sotto BASIC (col comando R seguito da Cr) e digitiamo, in modo immediato:

POKE$ 6BD3, $8F

Dopo di che digitiamo il comando List. Vedremo che l'istruzione precedente è diventata ora:

10 PRINT

Infatti, come avevamo già visto, il valore 8Fh era il codice associato all'istruzione Print.

Ora possiamo, con tante Poke all'indirizzo 6BD3h date in modo immediato, inserire dei valori esadecimali

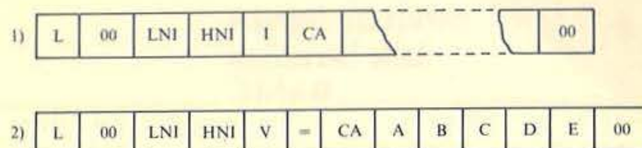


Figura 6. *Esempi di strutture interne:*
1) Istruzione,
2) Istruzione di assegnazione (Esempio V = N).
Dove:
L = lunghezza (in byte) dell'istruzione in memoria;
LNI = valore basso del numero di linea;
HNI = valore alto del numero di linea;
I = codifica dell'istruzione (vedere tabella 1);
CA = codice accompagnatorio. I valori accertati che può assumere sono i seguenti:
- *15h per numeri reali decimali,*
- *16h per numeri esadecimali,*
- *20h per le istruzioni,*

V = variabile.
Per le istruzioni di assegnazione (esempio V = N) avremo che: N = f(A,B,C,D,E)

Tabella 1. *Elenco istruzioni BASIC e relativa codifica esadecimale.*

Cod.	Istruz.	Cod.	Istruz.	Cod.	Istruz.	Cod.	Istruz.
80	Goto	A0	Poke	C0	Erase	E0	To
81	Gosub	A1	Off	C1	Error	E1	Step
82		A2	Mode	C2		E2	Then
83	Run	A3	Skip	C3	Usr	E3	Using
84	Return	A4	Plot	C4	Bye	E4	
85	Restore	A5	Line	C5		E5	
86	Resume	A6	Rline	C6		E6	Tab
87	List	A7	Move	C7	Def	E7	Spc
88		A8	Rmove	C8		E8	
89	Delete	A9	Tron	C9		E9	
8A	Renum	AA	Troff	CA		EA	
8B	Auto	AB	Inp#	CB		EB	Or
8C		AC		CC		EC	And
8D	For	AD	Get	CD		ED	
8E	Next	AE	Pcolor	CE	Wopen	EE	><
8F	Print	AF	Phome	CF	Close	EF	<>
90		B0	Hset	D0	Ropen	F0	=<
91	Input	B1	Gprint	D1		F1	<=
92		B2	Key	D2		F2	=>
93	If	B3	Axis	D3		F3	>=
94	Data	B4	Load	D4		F4	=
95	Read	B5	Save	D5		F5	>
96	Dim	B6	Merge	D6		F6	<
97	Rem	B7		D7		F7	+
98	End	B8	Console	D8		F8	—
99	Stop	B9		D9	Kill	F9	
9A	Cont	BA	Out#	DA		FA	
9B	Cls	BB	Circle	DB		FB	/
9C		BC	Test	DC		FC	★
9D	On	BD	Page	DD		FD	↑
9E	Let	BE		DE		FE	
9F	New	BF		DF		FF	

Struttura interna delle istruzioni BASIC

Tabella 2. *Elenco funzioni BASIC e relativa codifica esadecimale.*

Codice	Funzione	Codice	Funzione
80	INT	9E	JOY
81	ABS	A0	CHR$
82	SIN	A1	STR$
83	COS	A2	HEX$
84	TAN	AB	ASC
85	LN	AC	LEN
86	EXP	AD	VAL
87	SQR	B3	ERN
88	RND	B4	ERL
89	PEEK	B5	SIZE
8A	ATN	BA	LEFT$
8B	SGN	BB	RIGHT$
8C	LOG	BC	MID$
8E	PAI	C3	STRING$
8F	RAD	C4	TI$
95	EOF	C7	FN

compresi tra 00h e FFh, in modo da avere l'elenco di tutte le istruzioni BASIC presenti nell'interprete. Noteremo che da 00h a 1Fh non vedremo niente di interessante (tali codici sono utilizzati per definire costanti numeriche decimali o esadecimali), da 20h a 7Fh vedremo tutti i caratteri che hanno codice ASCII corrispondente a tali valori (dallo spazio al carattere↓), mentre dal valore 80h al FFh vedremo le istruzioni BASIC che abbiamo raggruppato nella tabella 1.
In tale elenco vediamo istruzioni che non sono state dichiarate nel manuale. Tra queste le seguenti:

ERASE - KILL - AND - OR

originano un "Syntax Error", mentre funzionano le seguenti:

Cls = pulizia del video;
Tron = inizia l'esecuzione a tracciamento (cioè stampa tutti i numeri di linea eseguiti);
Troff = disabilita l'esecuzione a tracciamento;
π contiene il valore 3,1415927 (PI greco).

Un procedimento analogo può essere seguito per determinare l'elenco delle funzioni trattate dall'interprete BASIC. Battiamo New e poi la seguente funzione:

10 INT (A/B)

Entriamo, come al solito, nel Monitor e vediamo cosa c'è dall'indirizzo 6BCFh in poi. Troveremo:

0C - 00 - 0A - 00 - FF - 80 - 28 - 41 - FB - 42 - 29 - 00

Dopo tutto quanto detto presumiamo che sia il valore 80h, all'indirizzo 6BD4h, il codice che identifica la funzione Int. Rientriamo nel BASIC e digitiamo, in modo immediato:

POKE$ 6BD4, $nn

con nn variabile da 00h, a FFh. Gli unici valori significativi che incontreremo sono evidenziati nella tabella 2.
Anche in questo caso incontriamo funzioni non elencate nel manuale. Di queste nuove funzioni le seguenti:

EOF - STRING

danno "Syntax Error" (forse anche perché, pur effettuando diversi tentativi, non abbiamo azzeccato la sintassi corretta), mentre le seguenti funzionano correttamente:
Joy = ritorna un valore che dipende dalla posizione assunta dalla barra di controllo giochi. Non abbiamo potuto provarla ulteriormente, perché sprovvisti di joystick;
Hex = ritorna il valore esadecimale che corrisponde al valore decimale dell'argomento.
In conclusione, come detto all'inizio, possiamo dire che l'analisi finora condotta, anche se incompleta, è stata decisamente utile, perché ci ha permesso di scoprire almeno 6 nuove istruzioni BASIC funzionanti. ■

PICCOLI ANNUNCI

Apple

Posseggo da breve tempo un Apple IIe. **Vorrei entrare in contatto con club o privati per consigli,** possibilmente in Piemonte. Carlo Audone - Via Marco Polo, 41 - 10129 Torino - Tel. 011/599587

Vorrei comunicare con ragazzi in possesso di un sistema Apple II da poco tempo. Scrivere soltanto. Roberto Vacchetti - Via Provinciale, 66 - 12050 Castelrotto (CN)

Vendo per Apple **un programma che calcola l'ammortamento** con i metodi francese, italiano, generico; creato in collaborazione con commercialisti. Per ulteriori informazioni rivolgersi a: Luigi Palumbo - Via A. Ristori, 8 - 00197 Roma - Tel. 06/802783

Vendo **software professionale sintesi sonoro-musicale** per Apple e utilities fra le migliori. Vendo inoltre stupendi games. Ubaldo Ponzio - Via Mazzini, 10 - 10127 Torino - Tel. 011/557672

Commodore

Per Commodore 64 realizzo su richiesta **programmi gestionali per rappresentanti, medici, amministratori** e per qualsiasi altra attività. Maurizio Galvini - Via Brunelleschi, 30/A - 20090 Trezzano S/N (MI) - Tel. 02/4455203

Causa cambio computer vendo **C64 più registratore, joystick e 15 giochi.** Materiale nuovo, ancora in garanzia, a L. 500.000 trattabili. Telefonare ore serali dalle 20 in poi. Daniele Allegri - Via P. Togliatti, 8 - 28017 S. Maurizio D'Opaglio (NO) - Tel. 0322/96283

Regalo 20 cassette con programmi di utilità e games, 2 libri con 50 listati per C64 **a chi acquista CBM 64, registratore C2N e stampante MPS802** a sole L. 1.000.000 non trattabile. Telefonare ore pasti. Paolo Panetta - Via Fionide, 1 sc. A - 74100 Taranto - Tel. 099/333144

Vendo **VIC 20 nuovissimo** più libro con 2 programmi e due riviste "Supervic" a L. 145.000. Giovanni Casandra - Corso Atellano, 100 - 81030 Sant'Arpino (CE) - Tel. 081/8918651

Anche per CBM 64 **contabilità generale, partita doppia,** programmi professionali per ingegneria con manuali, utility e gestionali. Giordano Bifolchi - Via Di Gracciano nel Corso, 111 - 53045 Montepulciano (SI) - Tel. 0578/757907

Affarone! Vendo **Commodore 64 più registratore C2N,** 4 supercartucce videogame, 1 cassetta videogame, joystick, manuale d'uso. Tutto l'occorrente per divertirsi a sole L. 490.000. Praticamente nuovo! Telefonare ore serali. Andrea Benedet - Via Lodovico Muratori - 20135 Milano - Tel. 02/7532935

Vendo **CBM 64 più registratore,** joystick, cassette con programmi e 2 manuali. Il tutto in ottime condizioni. Roberto Brambilla - Via Liguria, 16 - 20038 Seregno (MI) - Tel. 0362/230565

Sai programmare il tuo C64 in BASIC o in LM e abiti a Padova o dintorni? Se ti interessa scambiare idee o "trucchi", mettiti in contatto con me. Telefona ore serali. Tommaso Masi - Via Toscanini, 17 - 35031 Abano Terme (PD) - Tel. 049/810881

Vendo **VIC 20 completo di accessori** più registratore, 2 cartucce, 10 programmi gioco in cassetta, 3 libri sul VIC, garanzia e istruzioni in italiano a L. 350.000 spedizione compresa. Lorenzo Colombina - Via S. Agostino, 293 - 36100 Vicenza - Tel. 0444/568165

Vendo **VIC 20 più registratore C2N,** monitor fosfori verdi 10", espansioni 8 Kbyte e 3 Kbyte S.E., molto software a L. 550.000 trattabili. Telefonare ore pasti. Paolo Zaio - Via Del Castagnone, 33 - 15048 Valenza (AL) - Tel. 0131/951096

Vendo **Commodore 8032 più floppy 1 Mbyte,** stampante 160 cps con ampia libreria software. Usato pochissimo, prezzo da concordare. Telefonare ore pasti a: Marco Succi - Via Bartolini, 14 - 48100 Ravenna - Tel. 0544/38000

Cerco urgentemente una tastiera per VIC 20 a L. 70.000-80.000 al massimo e vendo 10 giochi per VIC 20 a L. 10.000 (cassetta compresa). Stefano Flammia - Via Fosso Cavallo, 91 - 83040 Fontanarosa (AV)

Vendo **cartucce nuove e sigillate** a L. 30.000 per VIC 20. Francesco Garaventa - Via San Gaetano, 14/A - 16030 Pieve Ligure (GE) - Tel. 010/3460432

Cerco **listati in linguaggio macchina** di qualsiasi tipo. Inoltre cerco il libro "Giochi con il 6502". Rispondo a tutti e assicuro il dovuto compenso. Roberto Rebecchi - Via Genova, 16 - 29029 Rivergaro (PC) - Tel. 0523/958252

A L. 720.000 vendo **CBM 64 più tape C2N** e 48 programmi. In omaggio dò: VIC 20, 50 programmi, 18 riviste, 1 joystick, 1 libro sul VIC, 2 cassette di giochi sempre per C64. Telefonare ore pasti. Marcello Rutter - Via Beata Giovanna, 63/A - 36061 Bassano del Grappa (VI) - Tel. 0424/26924

Vendo **VIC 20 più registratore dedicato C2N (tutto perfetto!)** più manuale in italiano, libro "Alla scoperta del VIC 20", 50 programmi, numerosi listati a L. 250.000. Inoltre regalo joystick. Massimo Bonaventura - Via Lorenzo Vidaschi, 31 - 00152 Roma - Tel. 06/534533

Vendo **CBM 3032 con registratore C2N e manuale** a L. 700.000. Telefonare dopo cena a: Roberto Biscani - Via Virgolana, 35 - 38057 Pergine Valsugana (TN) - Tel. 0461/532690

Vendo **Commodore VIC 20 con registratore C2N** e 55 giochi, un joystick, una paddle, libro BASIC 2 a prezzo eccezionale: L. 300.000 trattabili. Occasione, telefonate subito mi raccomando! Jacopo Rubbi - Via Paolo V - 44100 Ferrara - Tel. 0522/23953

Vendo **VIC 20 più espansione 16 Kbyte,** machine language code monitor, completo di registratore, cavetti e manuali per l'uso. Prezzo interessantissimo! Regalo inoltre una cassetta con più di 60 giochi (molti in LM). Massimo Pegoraro - Piazzetta S. Gaetano, 6 - 36015 Schio (VI) - Tel. 24749

Vendo **VIC 20 più cavi e trasformatore,** registratore, espansioni 16 e 8 Kbyte, motherboard due entrate, manuale, 100 programmi, in blocco. Preferibilmente in zona Milano. All'acquirente regalo joystick. Massimiliano Frosi - Via Magellano, 5 - 20094 Corsico (MI) - Tel. 02/4408498

Vendo **una cartridge più cassette** per Commodore 64 e VIC 20. Vendesi registratore della Commodore a L. 78.000. Telefonare ore 20-21. Simone Nulchis - Via Sforza, 7 - 54031 Avenza-Carrara (MS) - Tel. 58512

Vendo **modem completo di attacchi** per CBM 64, velocità 300 baud con istruzioni e disco operativo a L. 250.000. Eventuale interfaccia RS232 a L. 50.000. Giampiero Toso - Via dell'Anguilla, 10 - 33050 Latisana Località Aprilia (UD) - Tel. 0431/53000

Vendo **Commodore 1541** a L. 550.000; Commodore 64 più registratore a L. 550.000; circa 600 programmi a L. 950.000. Il tutto in ottimo stato. Massimo Niccolai - Via Edilizia, 21 bis/C - 29100 Piacenza

Vendo causa doppio regalo **espansione programmabile (3-8-16 Kbyte) mai usata,** valore effettivo L. 170.000. Non vendo ad un prezzo inferiore alle L. 145.000. Telefonare dopo le ore 20 o scrivere a: Marco Cavallari - Via Mazzini, 11 - 20054 Nova Milanese (MI) - Tel. 0362/42540

Vendo **VIC 20 più registratore,** joystick, 3 cartucce giochi, 10 cassette piene di giochi in alta risoluzione (Castel Dungeon, Lo Scalatore, ecc.), riviste ad un prezzo sbalorditivo. Telefonare ore pasti a: Maurizio Marras - Via Mascagni, 5 - 07029 Tempio Pausania - Tel. 079/630123

Vendo **VIC 20** più 5 cassette fra cui 3 adventure, manuale istruzioni, 2 libri di programmi, interfaccia per registratore a L. 200.000. Il tutto nuovo, solo 3 mesi di vita; telefonare ore 20-21. Daniele Luparello - Corso Unione Sovietica, 409 - 10135 Torino - Tel. 011/6193008

Sinclair

Cambio, vendo **ZX Spectrum 48 Kbyte più interfaccia joystick,** 1 joystick, box sonoro, stabilizzatore ed un sacco di programmi a L. 450.000. Cambio solo con CBM 64 più registratore in buone condizioni. Elio Marchi - Via S. Donato, 72 - 55100 Lucca - Tel. 0583/584651

Per Spectrum 48 Kbyte cerco **giochi Adventure con grafica,** completi di istruzioni o manuali. Inoltre cerco programma che traduca il BASIC in linguaggio macchina. Francesco Ghirotti - Via Perilli, 11 - 48100 Ravenna - Tel. 0544/422254

Vendo **stampante ZX printer e quattro rotoli di carta metallizzata** a L. 80.000. Inoltre vendo interfaccia programmabile joystick Tenkolek a L. 40.000. Telefonare a: Paolo Ballocci - Via San Gottardo, 75 - 20052 Monza (MI) - Tel. 039/367709

Vendo **ZX Spectrum 48 Kbyte più registratore,** interfaccia joystick, manuale in italiano, 200 programmi inglesi. Il tutto a L. 490.000. Inoltre vendo Intellevision con cartucce Burger Time, Soccer, ecc. a L. 300.000. Alfonso Mauro - Via Vi Settembre, 36 - 84100 Salerno (SA) - Tel. 089/321389

Vendo **Sinclair ZX Spectrum 48 Kbyte; 1 anno di vita, ottimo stato;** manuale d'uso in italiano, alimentatore con levetta reset, corredo di programmi vari (giochi, utility) a L. 500.000 trattabili. Telefonare ore pasti. Mauro Padovan - Via Bellini, 24 - 47039 Savignano S.R. (FO) - Tel. 0541/945790

Occasione! Vendo per Spectrum **favolosa interfaccia programmabile per joystick** a sole L. 35.000! Telefonate subito, dopo le 19.30 a: Alessandro Gozzo - Viale Zara, 63 - 20159 Milano - Tel. 02/682659

Vorrei comprare un personal computer, preferibilmente Sinclair Spectrum o altri del valore che si aggiri sulle L. 500.000. Scrivere o telefonare ore pasti. Gaetano Bennardo - Via Italia, 2 - 92026 Favara (AG) - Tel. 31329

Vendo **Spectrum 16 Kbyte usato pochissimo, 1 anno di vita,** imballo originale con supergaranzia, alimentatore e cavi, manuale italiano, cassette Horizon e Intruders a sole L. 220.000. Franco Di Molfetta - Via XX Settembre, 24 - 25070 Sabbio Chiese (BS) - Tel. 0365/85151

Vendo a L. 5.000 non trattabili **la completa traduzione del manuale del gioco "The Lords of Midnight"** per Spectrum, eseguita interamente e dattiloscritta. Riccardo Ghini - Via Ferrari, 181/4 - 17011 Albisola Capo (SV) - Tel. 019/40448

Vendo **ZX Spectrum perfettamente funzionante** completo di alimentatore, cavi, circa 150 programmi (giochi, matematici, ingegneria civile) al fantastico prezzo di L. 600.000. Gerardo Manni - Via Foppo, 20 - 25043 Breno (BS) - Tel. 0364/22724

Novità! Ora il tuo computer **può anche sviluppare le espressioni algebriche.** Programma più manuale L 33.000 per C64 - L. 26.000 per VIC 20 più espansione - L. 27.000 per Spectrum 48 Kbyte. Aldo Montaldo - Via A. Zupi, 19 - 87100 Cosenza

Vendo per Spectrum **eccezionale ed unico programma per oroscopi** con dettagliate istruzioni. Per l'utilizzo del programma vendo a richiesta tabelle effemeridi, case, tempo siderale. Maurizio Baldecchi - Via Manzoni, 31 - 50015 Grassina (FI) - Tel. 055/441668

Vendo **ZX Spectrum con manuali,** cavi, alimentatore in ottime condizioni e numerose cassette a L. 270.000 non trattabili. Pietro Iacoboni - Via G. Marconi, 9 - 60054 Fabriano (AN) - Tel. 0732/3002

Attenzione: vendo per Sinclair QL **un programma che aggiunge 35 fantastici comandi di geometria** e vari altri programmi per il QL su microcassetta a L. 40.000 più spese. Per informazioni scrivere a: Gianni Oliviero - Via Cerlungo, 29 - 25018 Montichiari (BS)

Vendo **Spectrum 48 Kbyte più ZX Printer (perfettamente funzionanti)** più 100 programmi su listato, 100 programmi su cassetta, 3 libri sullo Spectrum, tutto a L. 450.000 trattabili. Anacleto Furlan - Viale Papà Giovanni, 35/10 - 31015 Conegliano Veneto (TV) - Tel. 0438/32818

Vendo **ZX Spectrum 16 Kbyte** con fantastici giochi ed utility. A chi lo desiderasse aggiungo registratore. Il tutto vale più di L. 670.000; cedo per sole L. 460.000 trattabili. Telefonare al pomeriggio. Fabio Bonamigo - Via Della Stazione, 1 - 36022 Cassola (VI) - Tel. 0424/813029

Vendo **ZX Spectrum 48 Kbyte,** interfaccia Kempston più un joystick, presa monitor e tasto reset, più il miglior software esistente, il tutto a L. 395.000. Tratto preferibilmente in zona Milano. Telefonare ore pasti. Paolo Ballocci - Via San Gottardo, 75 - 20052 Monza - Tel. 039/367709

Vendo **ZX81 più 16 Kbyte,** cavetti per televisore, cavetti per registratore, software a L. 160.000 trattabilissimi. Per ulteriori informazioni telefonare o scrivere a: Antonio Marino - Via Spartaco, 65 - 71013 S.Giovanni Rotondo (FG) - Tel. 0882/856663

Vendo per Spectrum 48 Kbyte **programma per la riduzione dei sistemi Totocalcio** scritto in linguaggio macchina. Superveloce! Consente la riduzione statistica e/o condizionata a errore. Francesco Arculeo - Via V 58, 6 - 90125 Palermo - Tel. 091/444432

PICCOLI ANNUNCI

Vendo **Spectrum 48 Kbyte (4 mesi di vita)** più interfaccia 1, microdrive, joystick Magnum, interfaccia Kempston, cavetti, manuali italiani, 200 programmi, libro Sinclair Spectrum Assembler e Linguaggio Macchina ecc. a L. 700.000 trattabili. Fabio La Porta - Via Montuoro, 5 - 90145 Palermo - Tel. 091/57140

Texas

Vendo **Texas TI 99/4A completo di cavetti,** registratore Inno Hit, manuali, giornali e 50 giochi in BASIC. Prezzo da concordare. Telefonare a: Paolo Deserafino - Via Castello, 99 - 10123 Torino - Tel. 011/549035

Vendo per TI 99/4A **peripheral box, disk controller, disk drive, 32 Kbyte RAM;** tutto a L. 1.500.000 non trattabili. Tratto solo con la Lombardia. Telefonare solo la domenica. Carlo Aceti - Via I Maggio, 4 - 22050 Verderio Inferiore (CO) - Tel. 039/510715

Vendo **simulazione di volo in tempo reale** per TI 99 con Extended BASIC a L. 8.000; è un programma che sfrutta al limite le capacità del TI, possiede ben 10 strumenti. Aldo Tomaselli - Via Roma, 18 - 38059 Strigno (TN) - Tel. 762007

Vendo **TI 99/4A più joystick,** cavo registratore, libro didattico a L. 220.000; il tutto imballato. Vendo inoltre il modulo Extended a L. 110.000, moduli SSS (Calcio, Scacchi, Parsec, Carwar, Attack) a L. 20.000 l'uno. Arcangelo Jeker - Via Guastalla, 15 - 20122 Milano - Tel. 78257

Vendo **TI 99/4A nuovo** più 3 giochi in cartuccia (Parsec, Scacchi, Munch Man), cavo per registrazione, manuale d'uso a L. 200.000. Telefonare ore ufficio. Fabio Maggi - Via Togliatti, 48/50 - 20017 Mazzo di Rho (MI) - Tel. 9303208

Vendo **TI 99 più Extended BASIC, espansione 32 Kbyte,** joystick, 400 programmi, manuali, libri, modulatore PAL, alimentatore al favoloso prezzo di L. 600.000 o permuto con Spectrum 48 Kbyte più microdrive e interfaccia 1. Luca Sburlati - Via Lomellina, 18 - 10132 Torino - Tel. 011/834116

Compro per Texas TI 99/4A **interfaccia RS232 per peripheral box.** Scrivere a Raffaele Ferrigno - Via D'Isernia, 4 - 80122 Napoli - Tel. 081/681571

Vendo **TI Extended BASIC più box expansion,** peripheral card, expansion card 48 Kbyte, 3 libri sul TI 99/4A a L. 700.000 non trattabili. Telefonare ore pasti. Giancarlo Russo - Via Matteo Rossi, 17 - 84100 Salerno - Tel. 089/238377

Vendo **TI 99/4A, SSS Alpiner e Munchman** con imballaggi originali, cassetta piena di programmi a L. 270.000. Telefonare ore pasti a: Stefano Bartolozzi - Via Piave, 104 - 52025 Montevarchi (AR) - Tel. 055/983414

Occasionissima, causa passaggio sistema, vendo **nuovissimo TI 99/4A più alimentatore, modulatore di frequenza,** coppia originale joystick e cavetti registratore, inoltre, Solid State Software Video Chess (Scacchi), SSS Alpiner, Beginner's BASIC Tutor della Texas, materiale per imparare il BASIC. Fornisco, inoltre, cassette con software: Labirinto 3D, Bruco di Bemer, Caverna di Marte, Music Teacher, ecc., il tutto completo di imballo originale a sole L. 270.000 trattabili. Telefonare o scrivere a: Stefano Andreatta - Via Pisani, 9/c - 30173 Mestre (VE) - Tel. 041/956449

Cerco **Extended BASIC** per TI 99. Prezzo massimo L. 150.000. Telefonare ore pasti. Matteo Morini - Via Tempia, 12 - 13042 Cavagliò (VC) - Tel. 0161/96283

Cerco **Extended BASIC** per TI 99/4A. Inoltre compro o cambio software; dispongo di alcuni programmi inediti (Totocalcio, English Teacher). Marcello Bartolotta - Piazza V. Veneto, 3 - 90143 Palermo - Tel. 091/522532

Vendo **TI 99/4A più BASIC esteso,** sintetizzatore vocale, Speech Editor, Terminal Emulator 2, interfaccia consolle-registratore, coppia joystick, oltre 250 giochi (BASIC ed Extended), molti programmi applicativi e gestionali, vari manuali d'uso. Il tutto ha solo tre mesi di vita ed è in condizioni eccezionali. Vendo a L. 510.000 trattabili anziché L. 813.000. Daniele Cerreti - Via Tomba di Nerone, 23 (Cassia) - 00189 Roma - Tel. 06/3661311

Vendo **TI 99/4A completo di:** modulatore, alimentatore, manuale, cavo per registratore, cassetta "BASIC Tutor", joystick, moduli SSS "Videogiochi 1" e "Videogiochi 2", giochi, libri a sole L. 250.000. Rosario Incarbone - Via Francesco Acri, 3 - 87100 Cosenza - Tel. 0984/20492

Regalo 35 programmi in TI BASIC e oltre 100 listati a chi acquista **TI 99/4A con manuale, imballo originale,** 2 joystick, 2 libri da L. 32.000, modulatore TV, cavo collegamento per registratore, trasformatore. Il tutto a L. 200.000 trattabili! Telefonare ore serali a: Marco Scopelliti - Tel. 02/9840452

Compro per TI 99/4A **programmi di ogni tipo, soprattutto giochi.** Scrivere inviando la lista ed il prezzo a: Giuseppe Rossoni - Via 4 Novembre, 3 - 24049 Verdellino (BG) - Tel. 882229

Vendo **TI 99/4A con accessori** (modulatore TV, trasformatore 220V, cavo registratore), BASIC esteso, libro per apprendimento. Allego manuali originali e registratore Texas con vari programmi. Tutto a L. 450.000. Alfonso Battista - Via Dell'Arco di San Calisto, 22 - 00153 Roma

Vendo in blocco: **Texas TI 99/4A, registratore, joystick,** 3 moduli SSS, 7 cassette con programmi vari e 3 libri sulla programmazione a L. 290.000. Telefonare ore pasti. Andrea Stefani - Via Elba, 9 - 00015 Monterotondo (RM) - Tel. 06/9005962

Vendo **TI 99/4A in ottime condizioni completo di:** consolle, alimentatore, cavo registratore, Extended BASIC, numerosissimi ottimi programmi. Telefonare ore pasti serali. Tutto a L. 300.000 trattabili. Marco Ferri - Via Lago Maggiore, 13 - 20075 Lodi (MI) - Tel. 0371/51585

Vendo **TI 99/4A** a sole L. 300.000, con 6 splendide cartucce (Mangman, Calcio, Chisolm-Trail, Wumpus, A-maze-ing, Alpiner). Telefonare o scrivere a: Massimo Pinto - Via Domenico Fontana, 184 - 80131 Napoli - Tel. 081/251598

Data 99: una stupenda raccolta di 18 programmi inediti per il tuo Texas TI 99/4A interamente scritti in TI BASIC. In questo primo numero troverai programmi di svago, utilità e didattica. La cassetta più il manuale di spiegazioni ti verranno spediti in contrassegno a sole L. 20.000 (spedizione compresa). Si assicura tempestività e massima serietà. Per richiedere il primo numero: Data 99 - Via Arenella, 81 - 80128 Napoli - Tel. 081/24079

Varie

Vendo **Atari, due joystick,** cinque cassette: Pac Man, Combat, Asteroids, Defender, Space War; tutto in perfette condizioni a L. 250.000. Andrea Cero - Via G. Verdi, 3 - 32010 Dozza di Zoldo (BL) - Tel. 0437/78314

Vendo **consolle Intellevision,** 1 anno di vita, con 3 cassette: Sharp Shark, Triple Action, Vectron; tutto a L. 200.000. Telefonare dopo le ore 20. Vendo anche cassette per Spectrum a 16-48 Kbyte a L. 15.000. Angelo Bollati - Via Resegone, 8 - 20010 Canegrate (MI) - tel. 0331/400631

Vendo **VCS Atari, tutto compreso,** a L. 190.000, cassette a L. 20.000 cadauna. Telefonare ore pasti chiedendo di Alfonso o Gianfranco. Alfonso Di Giampietro - Via Dei Caccia, 11 - 28100 Novara - Tel. 0321/380425

Vendo **computer Olivetti M20, otto mesi di vita,** con espansione memoria 64 Kbyte RAM, con i seguenti programmi della casa: Multiplan, Oliword, Assembler, Calcolo sismico e molti altri a L. 6.000.000. Marco Zazzeri - Via Valzania, 25 - 48100 Portofuori (RA) - Tel. 0544/433557

Vendo al miglior offerente **scheda AIM 65 Rockwell funzionante,** completa di tutti gli accessori, anche del mobile e dei manuali (5), oppure cambio con periferiche per Spectrum 48 Kbyte. Francesco Ghirotti - Via Perilli, 11 - 48100 Ravenna - Tel. 0544/422254

Vendo **base Intellevision** più Frog Bog, Soccer, Tron II, Ice Trek, Burger Time, Poker in ottimo stato, il tutto completo di imballaggio, a L. 200.000. Sergio Castaldo - Parco San Paolo, 32 - 80126 Napoli - Tel. 081/7674796

Vendo **le seguenti schede per micro N.E.:** LX383 int. tast. easdec. a L. 80.000; LX384 tastiera esadec. a L. 85.000; LX385 int. registratore a L. 115.000, LX386 espansione 8 Kbyte a L. 110.000. Marco Cozzani - Piazzetta U. Cheirasco, 1 - 19100 La Spezia - Tel. 0187/508817

Vendo **Apple II plus compatibile con unità disco,** controller e monitor 12" fosfori verdi. Il tutto a L. 1.650.000. Stefano Goria - Via Accademia Albertina, 40 - 10123 Torino - Tel. 011/510173

Vendo **Sharp MZ-731 con registratore e stampante (a 4 colori) incorporati,** con 11 giochi e alcune riviste a L. 650.000. Il computer ha solo tre mesi di vita, con garanzia ancora valida. Enrico Villanova - Via Botteniga, 61 - 31100 Treviso (TV) - Tel. 0422/24211

Vendo **Atari VCS 2600** più 6 cassette (Decathlon, Enduro, Vanguard, Raiders Of The Lost Ark, Ms. Pacman, Space Invaders), comando fuoco rapido Pointmaster. Il tutto a L. 350.000 trattabili. Daniele Luparello - Corso Unione Sovietica, 409 - 10135 Torino - Tel. 011/6193008

Vendo **consolle Intellevision** più 13 cartucce. Telefonare ore serali. Alessandro Violani - Via D. Secchia, 2 - 40131 Bologna - Tel. 051/554317

Sei anche tu un utente Victor-Sirius? Scrivimi! Potremmo esserci utili a vicenda. Salvo Giudice - Via Pio X, 50 - 31031 Caerano S. Marco (TV) - Tel. 0423/858201

Olivetti M24, IBM PC ed IBM compatibili: **cerco possessori di questi sistemi per scambio di esperienze e software,** linguaggi BASIC, Pascal, Assembler. Mario Manuzzi - Via Serraglio, 37 - 47023 Cesena (FO) - Tel. 0547/21947

Vendo **computer portatile Philips P2000C nuovo,** video 9" 24x80 (verde), 64 Kbyte RAM, 2 floppy 5,25" da 640 Kbyte ciascuno, tastiera italiano, completo di: CP/M 2.2, Fortran, MSBASIC, Magro 80, Link 80, WordStar, Mailmerge, SpellStar, CalStar, InfoStar, dBase II, Pascal MT e numerosi altri pacchetti di utilità come assemblatori, editor, disassemblatori, debug, ecc. Il tutto a L. 3.000.000. Telefonare a Enzo, tel. 02/425581

IN BASIC s.n.c., Via Sezze 22, 04100 Latina, tel. 0773/487631 (Prof. Carlo Sintini). Programmi di ogni tipo; scompilatore-compilatore per S.B.; manuali; traduzioni. Inviare L. 3.000 per lista.

Vendo **Sharp MZ-731** (plotter e registratore incorporati) 64 Kbyte più programma Easy Data, cassetta S-BASIC e 10 giochi. Telefonare ore serali. Marco Carpani - Via Cacciapiatti, 10 - 28100 Novara - Tel. 0321/34713

Cassette originali: "Load'n'Run", "Soft", "Libreria Software", corso "input" agg. ed altro. Svendo con regali. Annuncio sempre valido, solo zona Palermo/Trapani. Francesco Burgio - Via Notarbartolo, 49 - 90100 Palermo - Tel. 091/266096

Vendo **Intellevision in ottime condizioni** più una cassetta a L. 150.000. Vendo anche Frog Bog, Calcio, Sci a L. 20.000; Demon Attack, Burgertime L. 30.000. Oppure vendo il tutto in blocco a L. 240.0000. Federico Ciampi - Via Vecchia Fiorentina, 64 - 56023 S. Lorenzo alle Corti (PI) - Tel. 050/770447

Vendo **programmi di ingegneria civile (telai, travi, ecc.) per Sharp PC 1500** adattabili ad altri computer. Sansone Carmelo - Camera 130 - Viale Resistenza, 15 - 27100 Pavia - Tel. 0382/301271

Cambio **programmi ed esperienze su HP71B.** Sono interessato ad applicazioni tecnico scientifiche. Sarei anche interessato alla formazione di un club pocket computer HP. Contattare urgentemente. Annuncio sempre valido. Giuseppe Torluccio - Via Selice, 42 - 40026 Imola (BO) - Tel. 0542/35409

Causa passaggio ad unità superiore vendo **Sharp MZ-721 usato pochissimo** con cassette giochi e programmi vari, a L. 620.000. Telefonare ore pasti. Gianfranco Acerbi - Via T. Vecellio, 112 - 35132 Padova - Tel. 049/605216

Vendo **Atari Computer System** più 7 cassette quasi nuove. Scrivere o telefonare a: Renato Belardinelli - Piazza S. Gallo, 1 - 60100 Ancona - Tel. 071/22876

Vendo **stampante Microline u80** 80 caratteri a L. 900.000 con imballaggio originale e connessioni. Poche ore di funzionamento (scrittura bidirezionale). Telefonare ore pasti. Cristina Curti - Via Campeggi, 2 - 27100 Pavia - Tel. 0382/57089 407089

Svendo **stampante Microline u80** a L. 800.000; drive 8" a L. 800.000; monitor Olivetti mod. Tes 601 antiriflesso a L. 400.000; il tutto usato veramente poco. Telefonare ore pasti. Davide Mannozzi - Via Cattaneo, 11 - 27100 Pavia - Tel. 0382/472984

Cerco **chiunque sia in possesso del nr. 2 e 4 di "Personal Software".** Sono interessato solo a poche fotocopie delle parti riguardanti software per il PET. Scrivetemi! Roberto Rebecchi - Via Genova, 16 - 29029 Rivergaro (PC)

Vendo **Personal Kid 100% Apple compatibile** 64 Kbyte, 2 drive 5" Slim, monitor colori 14" a L. 2.500.000 ancora in garanzia. Telefonare dopo le 19. Massimo Ciucci - Viale Famagosta, 24 - 20142 Milano - Tel. 02/816427

PERSONAL SOFTWARE LIBRI

Gruppo Editoriale Jackson

Vuoi ordinare dei libri? Spedisci questo tagliando a: Gruppo Editoriale Jackson Via Rosellini, 12 - 20124 Milano

Nome Cognome

Indirizzo N.

Cap. Città Provincia

Partita I.V.A. (indispensabile per le aziende)

Si richiede l'emissione della fattura ☐

Inviatemi i seguenti libri:

Codice Libro	Quantità	Codice Libro	Quantità	Codice Libro	Quantità	Codice Libro	Quantità

☐ Pagherò al postino il prezzo indicato + L. 3.000 per contributo fisso spese di spedizione.

☐ Allego assegno n° .... di L. ....

☐ Non abbonato ☐ Abbonato sconto ☐ l'Elettronica ☐ Elettronica Oggi ☐ Automazione Oggi ☐ Elektor ☐ Informatica Oggi
☐ Computerworld ☐ Bit ☐ Personal Software ☐ Strumenti Musicali ☐ Video Giochi ☐ Home computer
☐ Telecomunicazioni Oggi ☐ PC Magazine ☐ Personal O ☐ Compuscuola ☐ Super VIC ☐ Super Sync

Data .... Firma ....

PERSONAL SOFTWARE PICCOLI ANNUNCI

Sei un lettore di Personal Software e vuoi entrare in contatto con tutti gli altri lettori per comprare, cambiare o vendere il tuo sistema o qualche particolare accessorio? Spedisci questo tagliando a: Gruppo Editoriale Jackson - Via Rosellini, 12 - 20124 Milano

☐ COMPRO ☐ CAMBIO ☐ VENDO

☐ Annuncio gratuito ☐ Annuncio evidenziato (allegare L. 5.000). Non si accettano fotocopie o fogli volanti allegati al coupon.

Nome .... Cognome ....

Via .... N. .... C.A.P. ....

Città .... Prov. .... Tel. ....

UNA PUBBLICAZIONE
DEL GRUPPO EDITORIALE JACKSON

PERSONAL SOFTWARE

ANNO 4 N. 28 MAGGIO 1985

DIRETTORE RESPONSABILE: Giampietro Zanga

CAPO REDATTORE: Lorenzo Barrile

DIRETTORE TECNICO: Riccardo Paolillo

COORDINAMENTO TECNICO: Luca Zaninello

GRAFICA E IMPAGINAZIONE: Ugo Luzzati

SEGRETARIA DI REDAZIONE: Silvia Borgonovo

REDAZIONE USA: Gabriella Martino, Sergio Mello-Grand
G.E.J. Publishing Group Inc.

HANNO COLLABORATO A QUESTO NUMERO:
C. Cappelli, C. Aceti, C. Poma,
F. Farina, D. Zampariolo, S. Guarinelli,
S. Cerutti, L. Marras, A. Furlan,
M. Giacobazzi, A. Guida, M. Spero,
M. Sangiorgio

UFFICIO ABBONAMENTI: Tel. 6880951/2/3/4/5

CONSOCIATE ESTERE:
USA: G.E.J. Publishing Group, Inc. a Subsidiary of -
Gruppo Editoriale Jackson 1307 S. Mary Avenue -
94087 Sunnyvale (CA) Tel. (408) 7730103 - Telex: 49959972

GRAN BRETAGNA: G.E.J. Publishing L.t.d. 187 Oxford Street London
WIR1AJ Tel. (01) 4392931 - Telex: (051) 21248

PUBBLICITA': Concessionario per l'Italia e l'Estero
J. Advertising S.r.l. – Via Restelli, 5 - 20124 Milano
Tel. (02) 6882895-6882458-6880606 Telex: 316213 REINA I

PUBBLICITA' PER USA E CANADA:
International Media Marketing 16704 Marquardt Avenue,
P.O. Box 1217 Cerritos (CA) 90701 - (213) 926-9552

FOTOCOMPOSIZIONE: Lineacomp S.r.l.
Via Rosellini, 12 - 20124 Milano

STAMPA: Litografica S.r.l. (Busto Arsizio - MI)

Concessionario esclusivo per la diffusione in Italia e all'Estero: SODIP -
Via Zuretti, 25 - 20125 Milano

AUTORIZZAZIONE ALLA PUBBLICAZIONE:
Tribunale di Milano n. 69 del 20/2/1982

Spedizione in abbonamento postale Gruppo III/70
Prezzo della rivista L. 4.000. Numero arretrato L. 8.000
Abbonamento annuo (11 numeri) L. 34.000; per l'Estero L. 51.000
I versamenti vanno indirizzati a: Gruppo Editoriale Jackson – Via Rosellini, 12 - 20124 Milano - mediante emissione di assegno bancario, cartolina vaglia o utilizzando il C/C postale numero 11666203.
Per i cambi di indirizzo, indicare, oltre naturalmente al nuovo, anche l'indirizzo precedente, ed allegare alla comunicazione l'importo di L. 500, anche in francobolli.

Il Gruppo Editoriale Jackson è iscritto nel Registro nazionale della Stampa
al n. 117 - Vol. 2 - Foglio 129 in data 17-8-1982

GRUPPO EDITORIALE JACKSON S.r.l.

DIREZIONE, REDAZIONE, AMMINISTRAZIONE:
Via Rosellini, 12 - 20124 Milano - Telefoni: 68.80.951/2/3/4/5

SEDE LEGALE: Via G. Pozzone, 5 - 20121 Milano

DIREZIONE EDITORIALE: Daniele Comboni

DIREZIONE AMMINISTRATIVA: Giuliano Di Chiano

DIREZIONE DIVISIONE PERIODICI: Dario Tiengo

DIREZIONE DIVISIONE LIBRI
E GRANDI OPERE: Roberto Pancaldi

La biblioteca per ragazzi firmata Jackson.

Daniel Isaaman - Jenny Tyler

BATTAGLIE CON IL COMPUTER

Il gioco della guerra ha contagiato anche i computer. Qui vengono presentati vari giochi scritti in BASIC per "fare la guerra" sul proprio computer, che deve essere uno dei seguenti: BBC, Apple, Vic 20, TRS-80, ZX Spectrum e ZX 81.

Cod. 011D Pag. 48 Lire 9.000

Ian Graham

GIOCHI CON IL COMPUTER

Il libro, parla di giochi con il computer, una volta tanto visti dalla parte del computer, e non dell'utente.
In particolare spiega, in modo semplice ma preciso, come fa il computer a giocare, come fa a produrre immagini e suoni durante il gioco e come fa (in genere!) a vincere.

Cod. 006D Pag. 48 Lire 9.000

Brian Reffin Smith

IMPARIAMO A PROGRAMMARE

Il libro fornisce le conoscenze essenziali per incominciare a programmare in BASIC su di un personal computer.

Cod. 018D Pag. 48 Lire 9.000

PRIMI PASSI IN BASIC

Il libro propone l'apprendimento del linguaggio BASIC tramite lo studio e l'esame dettagliato di programmi già scritti.

Cod. 007D Pag. 48 Lire 9.000

Tony Potter - Ivor Guild

I ROBOT

Dopo una rapida panoramica su che cosa può fare un robot, il libro presenta una serie di robot con funzioni particolari: i robot a braccio, i robot operai, i robot nello spazio, etc.
Affronta poi il problema di come un robot può essere programmato e controllato da un computer, e di come praticamente si realizza un robot.

Cod. 003D Pag. 48 Lire 9.000

Judy Tatchell - Bill Bennett

CONOSCERE IL PERSONAL

Il libro spiega che cosa si può fare con un personal computer, come si usa e come funziona.

Cod. 008D Pag. 48 Lire 9.000

Lynn Miring - Ian Graham

RIVOLUZIONE INFORMATICA

Il volume è rivolto in particolare ai bambini, ma anche a tutti coloro che, presto o tardi, verranno coinvolti dalle nuove tecniche informatiche più come utenti che come operatori.

Cod. 004D Pag. 48 Lire 9.000

GRUPPO EDITORIALE JACKSON

ritagliare (o fotocopiare) e spedire in busta chiusa a:
GRUPPO EDITORIALE JACKSON - Divisione Libri - Via Rosellini, 12 - 20124 Milano

CEDOLA DI COMMISSIONE LIBRARIA

VOGLIATE SPEDIRMI

n° copie	codice	Prezzo unitario	Prezzo totale
		Totale	

☐ Pagherò contrassegno al postino il prezzo indicato più **L. 3.000 per contributo fisso spese di spedizione.**

Condizioni di pagamento con esenzione del contributo spese di spedizione:

☐ Allego assegno della Banca ____________

n° ____________

☐ Allego fotocopia del versamento su c/c n. 11666203 a voi intestato

☐ Allego fotocopia di versamento su vaglia postale a voi intestato

Nome ____________

Cognome ____________

Via ____________

Cap. ______ Città ______ Prov. ______

Data ______ Firma ______

Spazio riservato alle Aziende. Si richiede l'emissione di fattura

Partita I.V.A. | | | | | | | | | | | |

ORDINE MINIMO L. 50.000